

SKRIPSI

**PERANCANGAN PEMBUATAN *SLAVE COMPUTER CONTROL* UNTUK
PENGOPERASIAN *SMART BUILDING* JURUSAN TEKNIK ELEKTRO
ITN MALANG**



**Disusun Oleh :
EKO APRIYANTO
NIM : 03.17.109**



**JURUSAN TEKNIK ELEKTRO S - 1
KONSENTRASI ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
SEPTEMBER 2007**

LEMBAR PERSETUJUAN

PERANCANGAN PEMBUATAN *SLAVE COMPUTER CONTROL* UNTUK
PENGOPERASIAN *SMART BUILDING* JURUSAN TEKNIK ELEKTRO
ITN MALANG

SKRIPSI

*Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik
Pada Jurusan Teknik Elektro Strata Satu (S-1) Konsentrasi Elektronika*

Disusun Oleh :

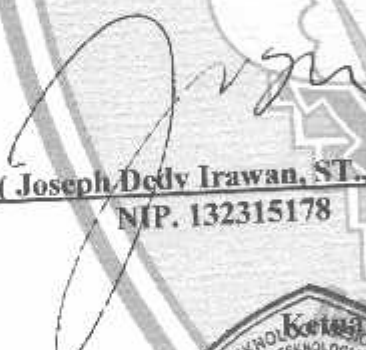
EKO APRIYANTO

NIM : 03.17.109

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

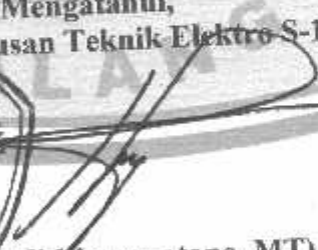

(Joseph Dedy Irawan, ST., MT.)
NIP. 132315178


(I Komang Somawirata, ST., MT.)
NIP.P. 1030100365

Mengatahui,

Ketua Jurusan Teknik Elektro S-1




(E. Audi Limpraptono, MT)
NIP.Y. 1039500274

JURUSAN TEKNIK ELEKTRO S - 1
KONSENTRASI ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKRO S-1
KONSENTRASI ELEKTRONIKA

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : Eko Apriyanto
NIM : 03 17 109
Masa Bimbingan : 05 JUNI 2007 s/d 05 DESEMBER 2007
Judul Skripsi : Perencanaan Pembuatan *Slave Computer Control* untuk
Pengoperasian *Smart Building* Jurusan Teknik Elektro
ITN Malang

Dipertahankan di hadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1)

pada :

Hari : Senin
Tanggal : 03 September 2007
Nilai : 84,25 (A) *Buy*

Panitia Majelis Penguji,



Ketua

(Ir. Muchtar Asrori, MSME.)
NIP.Y. 1018100036

Skertaris

(Ir. F. Yudi Limpraptono, MT.)
NIP.Y. 1039500274

Anggota Penguji,

Penguji 1

(Ir. Widodo Puji Mulyanto, MT.)
NIP.Y. 1028700171

Penguji 2

(Dr. Cahyo Crvsdian, Msc.)
NIP. 1030400412

BISMILLAHIRRAHMAANIRAHIM.....

*Alhamdulillah, Puji syukur Kehadirat Allah Subhanahu Wa Ta'ala
yang senantiasa mencurahkan Rahmat, Hidayah Ridlo dan Inayah-Nya,
Shalawat serta Salam tetap tercurah kepada junjungan
Nabi Besar Muhammad SAW...*

*Skripsi dan gelar sarjana Eko persembahkan buat Bapak dan Ibu
tercinta..*

*Yang telah berjuang dengan daya, upaya dan kasih sayangnya demi putranya
agar menjadi lebih baik di kehidupan ini...*

Semoga Allah selalu mengabulkan Do'a Bapak dan Ibu...

*Serta memberikan kesempatan kepada putranya agar selalu dapat
membahagiakan Bapak dan Ibu... amien...*

*dan semoga Allah selalu memberi kesehatan, kedamaian, ketentraman,
kebahagian dan umur panjang yang barokah... kepada Bapak dan Ibu...
amien...*

*Terimakasih pada Mbah, Pakpuhi, Bude, Paksek, Mas-mas, Adek Dwi dan
Seluruh keluarga besar ku yang ku sayangi atas segala bantuan moril dan
do'anya, Semoga Allah Membalasnya dengan balasan yang belipat-lipat dan
semoga Allah menjaga keharmonisan keluarga besar kita amien...*

*Matur Suwon Sanget damel Lek Joko sekeluarga, Lek Untung sekeluarga,
Ahmat dan saudara-saudaraku atas segala bantuannya... semoga Allah
mengabulkan Setiap hajat baik kita... amien...*

*Sukron Katsir buat Apis, Tiya, Budi, Inul dan Intan atas segala bantuan
moril spirituil, dukungan, nasehat dan masukukannya... semoga semuanya
mendapatkan kebahagiaan dunia dan akhirat... amien...*

*Matur Suwon Sanget damel Mas Mupit, Mbak Ayu, Mas Misbah, Adek Rif
n Azka serta keluarga (Bapak, Ibu n Mbah) atas segala dukungan, nasehat,
semangat, hiburan, bantuan moril spiritual... Mudah-mudahan Allah
memberikan kesehatan n kebahagiaan selalu untuk semua... amien...*

*Terimakasih pula buat crew Lab Elka Digital (Pakde, Mas Irawan, Bagus
dan semuanya) atas segala bantuannya...*

Terimakasih pula buat adek-adek ku, sahabat- sahabatku, teman-temanku (mohon maaf gak disebutin satu persatu), atas semua bantuannya untuk ku...
Semoga Allah selalu dekat dan memberikan hidayah-Nya pada kita, amien...

Yang terakhir terimakasih buat semua baik yang dengan sengaja, tidak sengaja, terpaksa, Ikhlas Jujur, minta Imbalan, Banyak Omong, diem, gak ngenaki, tapi dah Bantu Eko tuk nylesain kuliah n skripsi ini... Semoga Allah selalu dekat dan memberikan hidayah pada mereka, amien...

Ilmu yang barokah adalah ilmu yang meskipun sedikit tetapi diamalkan dan membawa manfaat pada yang memiliki dan orang disekitarnya...

Rasulullah bersabda, 'manusia yang paling utama ialah orang yang beriman yang berilmu, yang bila dibutuhkan ia berkenan memberikan manfaat dan jika tidak dibutuhkan akan bisa mencukupinya sendiri'

Orang pintar memang sangat diperlukan tetapi orang mengerti akan lebih Diperlukan...

Dalam menghadapi segala sesuatu
Yakinlah pada Allah dan yakinlah dengan apa yang ada pada diri kita
insyaAllah kita dapat melewatinya.

Jalani Segala sesuatu dengan Sabar dan Ikhlas
yakinlah semua itu datangnyanya hanya dari Allah, dan semua akan kembali pada-Nya.

ABSTRAKSI

JUDUL : PERANCANGAN PEMBUATAN *SLAVE COMPUTER CONTROL* UNTUK PENGOPERASIAN *SMART BUILDING* JURUSAN TEKNIK ELEKTRO ITN MALANG

Disusun Oleh : Eko Apriyanto, 03.17.109

**Dibawah Bimbingan : Dosen Pembimbing I – Joseph Dedy Irawan, ST., MT.
Dosen Pembimbing II – I Komang Somawirata, ST., MT.**

Kata Kunci : *Smart Building, TCP/IP, Socket, Delphi.*

Slave Computer ini dibuat untuk mengontrol dan memonitor *Computer Control* yang berada di dalam ruang kontrol dalam pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang. Sehingga keadaan setiap titik beban tetap dapat dimonitor dan dikontrol meskipun aktifitas kampus telah usai dan ruang kontrol telah tutup.

Prinsip kerja dari *Slave Computer* adalah memonitor dan mengontrol *Computer Control* dalam pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang. *Slave Computer* berkomunikasi dengan *Computer Control* menggunakan jaringan TCP/IP dengan memanfaatkan aplikasi *Socket*. *Slave Computer* dapat bekerja bila *Computer Control* dalam keadaan aktif. Pada saat program dijalankan *Slave Computer* akan meminta data status *Panel* pada *Computer Control*, dan *Computer Control* akan memberikan data status *Panel* sesuai dengan yang diminta oleh *Slave Computer*. Jika *Slave Computer* menginginkan perubahan status *Panel*, *Slave* akan memerintahkan *Computer Control* untuk melakukan perubahan status *Panel*.

Dari pengujian didapatkan respon perubahan status *Panel* atas permintaan *Slave Computer* dapat berlangsung rata-rata antara 1,4 s/d 1,5 detik. Jika dalam waktu tersebut tidak ada perubahan status pada *Panel* dapat berarti bahwa data yang dikirim tidak diterima dengan lengkap oleh *Panel*.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah Subhanahu Wa Ta'ala yang senantiasa mencurahkan Rahmad, Hidayah, Ridlo dan Inayah-Nya, serta anugerah kesahatan kepada penyusun sehingga skripsi yang berjudul "Perancangan Pembuatan *Slave Computer Control* Untuk Pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang" ini dapat diselesaikan. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro Strata Satu (S-1) Institut Teknologi Nasional Malang.

Penyelesaian skripsi dan penyusunan laporan ini tidak lepas dari saran, dukungan serta bantuan dari berbagai pihak untuk mencapai kesempurnaan. Tapi penyusun menyadari, kesempurnaan hanya Milik Allah SWT dan kita selaku makhluk hanya bisa berbuat yang terbaik semampu kita. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT., selaku Ketua Jurusan Teknik Elektro S-1.
2. Bapak Joseph Dedy Irawan, ST., MT., selaku Dosen Pembimbing 1.
3. Bapak I Komang Somawirata, ST., MT., selaku Dosen Pembimbing 2.
4. Bapak, Ibu Dosen serta seluruh Staf Jurusan Teknik Elektro S-1.
5. Bapak H. Sabarudin dan Ibu Hj. Siti Nurhasanah serta keluarga yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
6. Rekan Instruktur di Seluruh Laboratorium Jurusan Elektronika, dan
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Harapan penyusun semoga laporan skripsi ini dapat memberikan manfaat.

Malang, Agustus 2007

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii

BAB I : PENDAHULUAN

1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Rumusan Masalah	3
1.4. Batasan Masalah	3
1.5. Metodologi	3
1.6. Sistematika Penulisan	4

BAB II : TEORI DASAR

2.1. Pendahuluan	5
2.2. Mikrokontroller AT89S51	5
2.2.1. Struktur Memory AT89S51	9
2.2.2. RAM Internal	11
2.2.3. Register Fungsi Khusus (<i>Sepecial Function Register</i>)	12
2.2.4. <i>Timer</i> dan <i>Counter</i> dalam MCS51	16
2.2.5. Komonikasi Data Serial	18

2.3. Memori Eksternal	20
2.4. <i>Interface Unit</i> RS-232	23
2.4.1. Karakteristik Elektris	23
2.4.2. Karakteristik Mekanis	24
2.5. Komonikasi Serial 485	26
2.5.1. RS-485	26
2.5.2. Karakteristik Elektris.....	27
2.5.3. Karakteristik Mekanis	28
2.5.4. Komonikasi Multi Point dengan RS-485	29
2.6. Relay	31
2.7. Susor Cahaya (Luxtron LX-1102)	33
2.8. Model Refensi TCP/IP.....	34
2.8.1. Karakteristik Lapisan TCP/IP.....	35
2.8.2. Lapisan-Lapisan Model TCP/IP	35
2.8.2.1. Lapisan Aplikasi.....	36
2.8.2.2. Lapisan Transportasi (Transport).....	37
2.8.2.3. Lapisan Internet.....	40
2.8.2.4. Lapisan Antar muka Jaringan/Fisik	44
2.8.3. Transmisi Data pada Model TCP/IP.....	44
2.9. Borland Delphi 7.0	45
2.9.1. IDE Delphi.....	46
2.9.2. Menu Borland Delphi.....	49
2.9.3. Komponen Visual dan Nonvosual.....	51
2.9.4. <i>Socket</i>	51

2.9.4.1. Istila-Istilah	52
2.9.4.2. Komponen-Komponen <i>Socket</i>	55
2.9.4.3. Koneksi <i>Socket</i>	56
2.9.4.2. <i>Socket</i> untuk Pengiriman dan Penerimaan Data.....	57

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan.....	58
3.1.1. Blok Diagram Keseluruhan Sistem	58
3.2. Prinsip Kerja Alat.....	60
3.3. Perangkat Keras (Unit Panel dan Sensor yang ada pada Smart Building Jurusan Teknik Elktro ITN Malang)	61
3.3.1. Blok Diagram Unit Switching Panel	61
3.3.2. Komunikasi Data antara Computer control dan Panel	64
3.3.3. Sensor Cahaya (LX-1102).....	67
3.4. Perancangan Perangkat Lunak (Software).....	68
3.4.1. Program Aplikasi Komputer.....	68
3.4.2. FlowChart.....	69
3.4.3. Perancangan Pembuatan Software Menggunakan Bahasa Pemrograman Borland Delphi7.....	71
3.4.3.1. Penyusunan Desain Software.....	71

BAB IV PENGUJIAN RANGKAIAN DAN ANALISA DATA

4.1. Pengujian Pengiriman Data dari Slave Computer ke Computer Control.....	133
4.1.1. Tujuan.....	133

4.1.2. Peralatan Yang Digunakan.....	134
4.1.3. Langkah Pengujian.....	134
4.1.4. Hasil Pengujian.....	135
4.2. Pengujian Jawaban dari Computer Control atas Pertanyaan Slave Computer dengan Membandingkan Jawaban Panel atas Pertanyaan Computer Control.....	136
4.2.1. Tujuan.....	136
4.2.2. Peralatan yang Digunakan.....	136
4.2.3. Langkah Pengujian.....	137
4.2.4. Hasil Pengujian.....	138
4.3. Perubahan Status pada Computer Control dan Panel atas Permintaan Slave Computer dan mengetahui respon terhadap perubahan status.....	139
4.3.1. Tujuan.....	139
4.3.2. Peralatan.....	139
4.3.3. Langkah Pengujian.....	139
4.3.4. Hasil Pengujian.....	141

BAB V PENUTUP

5.1. Kesimpulan	156
5.2. Saran	156

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Susunan Pin dari AT89S51	7
Gambar 2.2 Alamat RAM Internal dan <i>Flash</i> PEROM.....	10
Gambar 2.3 Peta Memory dan Special Fungsi Register.....	12
Gambar 2.4 Konsep Dasar Timer/Counter sebagai sarana input	17
Gambar 2.5 EEPROM AT24C08.....	20
Gambar 2.6 Device Address	22
Gambar 2.7 Format Penulisan Data	22
Gambar 2.8 Format Pembacaan Data.....	18
Gambar 2.9 Format Pembacaan Data dengan Alamat Acak.....	23
Gambar 2.10 Level Logika Standar RS-232	23
Gambar 2.11 Konfigurasi Pin Konektor RS-232	24
Gambar 2.12 Level Tegangan Saluran Penerima.....	27
Gambar 2.13 Bentuk Mekanis Komponen IC RS-485.....	28
Gambar 2.14 Komunikasi 2 <i>Ware</i>	29
Gambar 2.15 Komunikasi 4 <i>Ware</i>	31
Gambar 2.16 Cara Kerja Relay	32
Gambar 2.17 Relay SPST.....	33
Gambar 2.18 Relay SPDT.....	33
Gambar 2.19 Relay SPDT	33
Gambar 2.20 Lapisan TCP/IP	36
Gambar 2.21 Lembar Kerja Delphi.....	46
Gambar 2.22 Tampilan Komponen <i>ClientSocket</i>	55

Gambar 2.23 Tampilan Komponen <i>ServerSocket</i>	55
Gambar 3.1. Diagram Blok Keseluruhan Sistem	58
Gambar 3.2. Blok Diagram Panel Kontrol	62
Gambar 3.3 Blok diagram unit switching panel	64
Gambar 3.4 Format Data Komunikasi Serial	65
Gambar 3.5. Diagram Alir Program Computer Controll	69
Gambar 3.6. Diagram Alir Program Slave Komputer	70
Gambar 3.7. Menyusun Tampilan Form Desain untuk Koneksi dengan Computer Control	72
Gambar 3.8. Menyusun Tampilan Form Desain untuk Penyetingan	75
Gambar 3.9. Desain Tampilan Halaman lantai satu	83
Gambar 3.10. Desain Tampilan Halaman lantai Dua	100
Gambar 3.11. Desain Tampilan Halaman Lantai Tiga	111
Gambar 3.12. Desain Tampilan Halaman Lantai Empat	122
Gambar 4.1 Diagram Blok Pengujian Pengiriman Data	134
Gambar 4.2 Data yang Dikirim dan Diterima Oleh Slave Computer	135
Gambar 4.3 Data yang Dikirim dan Diterima Oleh Computer Control	136
Gambar 4.4 Diagram Blok Pengujian Jawaban	137
Gambar 4.5 Diagram Blok Perintah	140
Gambar 4.6 Tampilan pada Slave Computer Sebelum melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu	141
Gambar 4.7 Tampilan pada Computer Control Sebelum melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu	141

Gambar 4.8	Tampilan pada Slave Computer Setelah melakukan penekanan	
	Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu	142
Gambar 4.9	Tampilan pada Computer Control Setelah melakukan penekanan	
	Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu	142
Gambar 4.10	Tampilan pada Slave Computer Setelah melakukan penekanan	
	Tombol Lab. Elektronika Analog untuk Pemadaman Lampu	143
Gambar 4.11	Tampilan pada Computer Control Setelah Melakukan Penekanan	
	Tombol Lab. Elektronika Analog untuk Pemadaman Lampu	143
Gambar 4.12	Tampilan pada Slave Computer Sebelum melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu ...	146
Gambar 4.13	Tampilan pada Computer Control Sebelum melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu ...	146
Gambar 4.14	Tampilan pada Slave Computer Setelah melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu ...	147
Gambar 4.15	Tampilan pada Computer Control Setelah melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu ...	147
Gambar 4.16	Tampilan pada Slave Computer Setelah melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pemadaman Lampu ...	148
Gambar 4.17	Tampilan pada Computer Control Setelah melakukan penekanan	
	Tombol Kalab. Elektronika Analog untuk Pemadaman Lampu ...	148
Gambar 4.18	Tampilan pada Slave Computer Sebelum Melakukan Penekanan	
	Tombol Lab. Elektronika Digital untuk Pengaktifaan Lampu	151
Gambar 4.19	Tampilan pada Computer Control Sebelum Melakukan Penekanan	
	Tombol Lab. Elektronika Digital untuk Pengaktifaan Lampu	151

Gambar 4.20 Tampilan pada Slave Computer Setelah melakukan penekanan Tombol Lab. Elektronika Digital untuk Pengaktifaan Lampu	152
Gambar 4.21 Tampilan pada Computer Control Setelah Melakukan Penekanan Tombol Lab. Elektronika Digital untuk Pengaktifaan Lampu	152
Gambar 4.22 Tampilan pada Slave Computer Setelah melakukan penekanan Tombol Lab. Elektronika Digital untuk Pemadaman Lampu.....	153
Gambar 4.23 Tampilan pada Computer Control Setelah melakukan penekanan Tombol Lab. Elektronika Digital untuk Pemadaman Lampu.....	153

DAFTAR TABEL

Tabel 2.1 Tabel Keluarga MCS-51	6
Tabel 2.2 Tabel Fungsi-Fungsi Khusus Port 3	8
Tabel 2.3 Fungsi Pin-Pin dari EEPROM AT24C08.....	21
Tabel 2.4 Fungsi Pin dalam DB9	25
Tabel 2.5 Spesifikasi RS-232	26
Tabel 2.6 Spesifikasi RS-485	28
Tabel 3.1 Kode Panel	66
Tabel 3.2 Merubah isi Properti pada Layar Connect ke Control.....	72
Tabel 3.3 Merubah isi Properti pada Layar Setting.....	76
Tabel 3.4 Merubah isi Properti dan Memasukan Event pada Halaman Lantai 1.	83
Tabel 3.5 Merubah isi Properti dan Memasukan Event pada Halaman Lantai 2.....	101
Tabel 3.6 Merubah isi Properti dan Memasukan Event pada Halaman Lantai 3	111
Tabel 3.7 Merubah isi Properti dan Memasukan Event pada Halaman Lantai 4	123
Tabel 4.1 Perbandingan Hasil Pengujian Data yang Diterima Computer Control dari Panel dan Data yang diterima Slave Computer dari Computer Control pada saat Status Panel tidak ada Perubahan.	138
Tabel 4.2 Respon Perubahan Status Lampu pada Computer Control atas Permintaan Slave Computer Dengan Penekanan Tombol Lab. Elektronika Analog	144

Tabel 4.3	Respon Perubahan Status Panel atas Permintaan Slave Computer	
	Dengan Penekanan Tombol Lab. Elektronika Analog	145
Tabel 4.4	Respon Perubahan Status Lampu Pada Computer Control atas	
	Permintaan Slave Computer Dengan Penekanan Tombol	
	Kalab. Elektronika Analog	149
Tabel 4.5	Respon Perubahan Status Panel atas Permintaan Slave Computer	
	dengan Penekanan Tombol Kalab. Elektronika Analog	150
Tabel 4.6	Respon Perubahan Status Panel atas Permintaan Slave Computer	
	Dengan Penekanan Tombol Lab. Elektronika Digital	154
Tabel 4.7	Respon Perubahan Status Panel atas Permintaan Slave Computer	
	Dengan Penekanan Tombol Lab. Elektronika Digital	155



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi di segala bidang, teknologi elektronika juga mengalami kemajuan yang sangat pesat sehingga hampir setiap peralatan yang digunakan manusia tidak lepas dari penggunaan berbagai macam peralatan elektronika yang ada. Pada perkembangannya, teknologi elektronika menuntut manusia untuk menciptakan suatu peralatan elektronika yang tepat guna dan praktis sehingga dapat membantu semua kegiatan menjadi lebih mudah. Penggunaan teknologi elektronika sebagai unit-unit pengendali sudah sangat luas, hal ini dikarenakan peralatan-peralatan yang dikontrol secara elektronik lebih banyak memberikan kelebihan dan kemudahan dalam penggunaannya, seperti dapat melakukan pengontrolan secara otomatis.

Demikian juga dengan bangunan gedung, saat ini bangunan gedung modern tidak lepas juga dari peralatan elektronik. Istilah *building automation system* kerap kali ditemukan pada gedung-gedung modern.

Pada sebuah gedung yang sangat besar, sering dijumpai masalah yang berkaitan dengan energi listrik. Sebagai contoh pada gedung-gedung perkantoran sering dijumpai kasus kelupaan mematikan *air condition*, lupa mematikan atau menyalakan lampu, sehingga membawa efek pemborosan energi. Dan seperti yang kita rasakan saat ini, energi listrik sudah semakin langka dan sangat mahal. Sehingga perlu mencari solusi untuk proses penghematan energi listrik tersebut.

Langkah untuk penghematan energi tersebut pada gedung JURUSAN TEKNIK ELEKTRO ITN MALANG telah dirancang sebuah system yang berfungsi untuk mengontrol seluruh beban listrik yang ada didalam gedung, sehingga setiap titik beban dapat dimonitor oleh operator bahkan dapat dijadwal keaktifannya.

Pada gedung Jurusan Teknik Elektro ITN Malang setiap titik beban dimonitor di dalam ruang kontrol, Sedangkan pada saat malam hari atau pada saat aktivitas kampus selesai ruang kontrol ditutup. Sehingga Setiap titik beban tidak lagi dapat dimonitor. Selama ini, pada saat ruang kontrol ditutup yang bertugas memonitor setiap titik beban adalah satpam dengan memeriksa secara manual.

Dari latar belakang diatas muncul pemikiran untuk merancang sebuah system yang berfungsi untuk mengontrol setiap titik beban dengan menggunakan *slave* dari *computer control* yang dapat di letakan pada ruang satpam dan dapat dioperasikan pada saat ruang kontrol ditutup. Sehingga tidak perlu lagi setiap titik beban dikontrol dengan cara manual.

1.2 Tujuan

Tujuan dari tugas akhir ini adalah membuat *Slave* dari *Computer Control* yang terhubung pada panel-panel *Smart Building* Jurusan Teknik Elektro ITN Malang, sehingga kondisi setiap titik beban tetap dapat dimonitor meskipun ruang kontrol telah tutup.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan pada bagian sebelumnya, maka dapat dirumuskan masalah yang akan dibahas yaitu bagaimana merancang dan membuat software yang dapat mengontrol *Computer Control* dalam pengoperasian *smart building* Jurusan Teknik Elektro ITN Malang.

1.4 Batasan Masalah

Dalam menyusun tugas akhir ini diperlukan suatu batasan masalah agar tidak menyimpang dari ruang lingkup yang akan dibahas. Adapun batasan masalahnya adalah sebagai berikut :

- a) Untuk mengendalikan *Computer Control* Menggunakan Bahasa Pemrograman Borland Delphi-7
- b) Komunikasi antara *Slave* dengan *Computer Control* menggunakan TCP/IP.
- c) Tidak membahas pembuatan *Hardware* pada *Smart Building* Jurusan Teknik Elektro ITN Malang
- d) *Slave Computer* bekerja pada saat *Computer Control* dalam keadaan aktif.

1.5 Metodologi

Metodologi yang dipakai dalam pembuatan skripsi ini adalah:

1. Studi Literatur

Dengan mencari referensi-referensi yang berhubungan dengan perencanaan pembuatan perangkat yang akan dibuat.

2. Field Research

Dengan melakukan penelitian secara langsung mengenai objek-objek yang berhubungan langsung dengan perencanaan perangkat yang akan dibuat.

3. *Design* dan Pembuatan perangkat

Yaitu meliputi pembuatan Sofwere pada program borland Dclphi-7.

4. Pengujian Perangkat

Dengan melakukan pengujian perblok perangkat dan kerja seluruh sistem pada perangkat tersebut.

1.6 Sistematika

Penulisan skripsi ini terbagi menjadi lima bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Membahas tentang latar belakang, tujuan, rumusan masalah, batasan masalah, metodologi dan sistematika penulisan pada penulisan skripsi ini.

BAB II LANDASAN TEORI

Berisikan tentang penjelasan dan teori-teori yang berhubungan dengan komponen-komponen yang digunakan dalam perancangan perangkat.

BAB III PERENCANAAN DAN PEMBUATAN PERANGKAT

Membahas tentang perancangan perangkat lunak.

BAB IV PENGUJIAN PERANGKAT

Membahas tentang pengujian peralatan secara keseluruhan dan analisa hasil pengujian.

BAB V PENUTUP

Berisikan kesimpulan yang didapat selama perancangan dan pembuatan Perangkat serta saran-saran.

BAB II

LANDASAN TEORI

2.1 Pendahuluan

Pada bab ini akan dibahas mengenai teori penunjang dari peralatan yang direncanakan. Teori penunjang ini akan membahas tentang komponen dan peralatan pendukung pada alat yang dibuat. Pokok pembahasan pada bab ini adalah :

- Mikrokontroller AT89S51
- EEPROM AT24C08
- RS-232
- RS-485
- Relay
- Sensor Cahaya (Lutron LX-1102)
- TCP/IP
- Borland Delphi-7 (*Socket*)

2.2 Mikrokontroller AT 89S51

Mikrokontroler AT89S51 merupakan salah satu rangkaian terintegrasi (*Intregated Circuits*) mikrokontroler yang tergabung dalam keluarga MCS-51.

Beberapa fitur yang terdapat pada mikrokontroler ini ialah sebagai berikut :

1. 4 buah I/O (*input/output*) port 8 bit.
2. 4 Kbyte memori program (*Flash EEPROM*).

3. 128 byte RAM internal.
4. 2 buah *timer* 16 bit.
5. 64 kbyte maksimum eksternal *memory address*.
6. 210 bit *addressable memory location*.
7. Terdapat *interface* untuk komunikasi serial.

Keluarga IC MCS-51 memiliki beberapa jenis IC mikrokontroler seperti yang ditunjukkan pada tabel 2.1 berikut :

Tabel 2.1 Tabel Keluarga MCS-51^[1]

Nomor seri	Memori program internal(Flash EEPROM)	Memori data internal (RAM)	Timer
8051	4 k byte ROM	128 bytes	2
8031	-	128 bytes	2
8751	4 k byte EPROM	128 bytes	2
8052	8 kbyte Rom	256 bytes	3
8032	-	256 bytes	3
8752	8 kbyte EPROM	256 bytes	3
89c51	4 k byte EEPROM	128 bytes	2

Susunan pin pada mikrokontroler AT89S51 adalah sebagai berikut :



Gambar 2.1 Susunan pin dari AT89S51^[1]

Keterangan Pin mikrokontroler AT89S51

1. Port 0 : Port dua fungsi.

Dalam perancangan konvensional Port ini hanya digunakan sebagai port I/O serbaguna, namun dalam sistem yang melibatkan memori eksternal port ini dapat digunakan secara bergantian sebagai port data maupun alamat (A0-A7).

2. Port 1 : Port I/O serba guna

3. Port 2 : Port dua fungsi.

Fungsinya sama dengan port 0, namun pada penggunaan memori eksternal, port ini digunakan untuk *bus* alamat *high* (A8-A15).

4. Port 3 : Port dua fungsi.

Selain dapat digunakan sebagai port I/O serbaguna dapat pula digunakan untuk fungsi-fungsi khusus lainnya seperti yang ditunjukkan pada tabel 2.2 berikut:

Tabel 2.2 Tabel Fungsi-fungsi khusus port 3 ^[1]

Bit	Nama pin	Alamat bit	Fungsi alternatif
P3.0	RXD	B0H	Comm serial receiver
P3.1	TXD	B1H	Comm serial tranceiver
P3.2	INT 0	B2H	Eksternal interrupt 0
P3.3	INT 1	B3H	Eksternal interrupt 1
P3.4	T 0	B4H	Eksternal Timer /Counter Input 0
P3.5	T 1	B5H	Eksternal Timer /Counter Input 1
P3.6	WR	B6H	Eksternal memori write enable
P3.7	RD	B7H	Eksternal memori read enable

5. PSEN (*Program store enable*)

Merupakan sebuah sinyal *output* yang berfungsi sebagai kontrol dalam membaca program maupun kode dari memori eksternal. Biasanya pin ini dihubungkan dengan *output enable* dari EPROM. Jika program yang dieksekusi berasal dari memori internal maka Pin ini akan berada pada kondisi tidak aktif (*high*).

6. ALE (*Address Latch Enable*)

Sinyal *output* ALE berfungsi dalam memisahkan waktu pengiriman data dengan alamat dalam penggunaan memori eksternal. Sinyal ALE membangkitkan pulsa sebesar 1/6 frekuensi oscilator. ALE hanya akan aktif pada saat mengakses memori eksternal.

7. EA (Eksternal akses)

Jika Pin EA diberi logika rendah maka mikrokontroler akan mengakses memori program eksternal. Sebaliknya jika Pin EA diberi logika tinggi, mikrokontroler akan mengakses memori program internal. Pin ini juga dipakai sebagai tegangan pemrograman EPROM atau *Flash Memori Internal*.

8. RST (Reset)

Pin ini berfungsi untuk mereset logika program dari awal. Biasanya untuk penggunaan *Power on reset*, sebuah resistor *pul-lup* ke VCC dan sebuah kapasitor yang terhubung ke *ground* dihubungkan pada pin ini.

9. Oscillator

Oscillator pada mikrokontroler ini diatur oleh besarnya nilai X'tal yang terhubung pada pin 18 dan pin 19. Besar nilai X'tal sekitar 12 Mhz.

10. Power

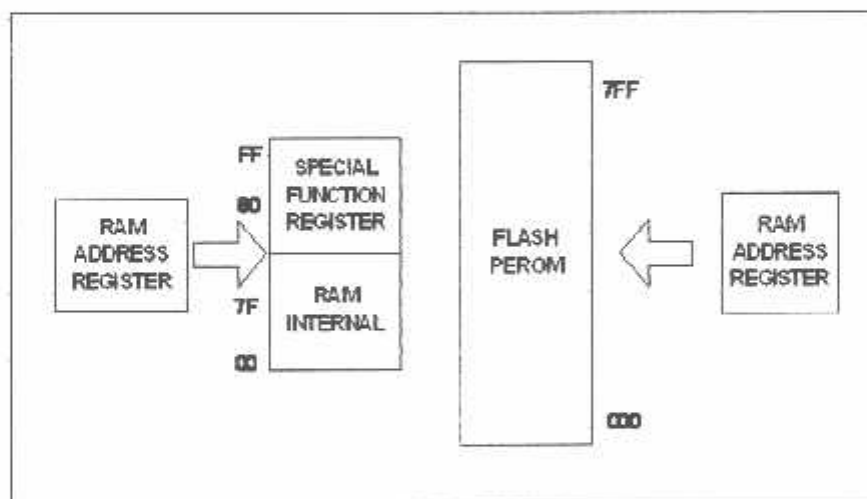
AT89S51 memerlukan tegangan catu sebesar 5 V DC pada pin 40 dan 0 V (*ground*) sebagai titik acuan pada pin 20.

2.2.1. Struktur Memory AT 89S51

Memori yang ada pada AT89S51 terdiri atas : (1) RAM Internal, memori sebesar 128 byte biasa difunakan untuk menyimpan variable atau data yang bersifat sementara. (2) *Special Function Register*(Register Fungsi Khusus), memori ini berisi register-register yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut seperti timer, serial dan

lain-lain. (3) *Flash* PEROM, memori yang digunakan untuk menyimpan instruksi-instruksi dalam AT89S51.

Dalam bukunya Paulus Andi Nalwan (2003) menyatakan: “AT89S51 mempunyai struktur memori yang terpisah antara RAM internal dan *Flash* PEROM-nya”. RAM Internal dialamati oleh *RAM Address Register* (Register Alamat RAM) sedangkan *Flash* PEROM dialamati oleh *Program Address Register* (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, maka walaupun RAM Internal dan *Flash* PEROM memiliki alamat yang sama yaitu alamat 00, namun secara fisiknya kedua memori tersebut tidak saling berhubungan. Untuk lebih jelasnya dapat dilihat pada gambar 2.2.



Gambar. 2.2 Alamat RAM Internal dan *Flash* PEROM^[2]

2.2.2. RAM Internal

RAM Internal pada AT89S51 terdiri atas :

1. Register Banks

AT89S51 mempunyai delapan buah register , terdiri dari R0-R7. Kedelapan buah register ini selalu terletak pada alamat 00H-07H saat sistem direset.

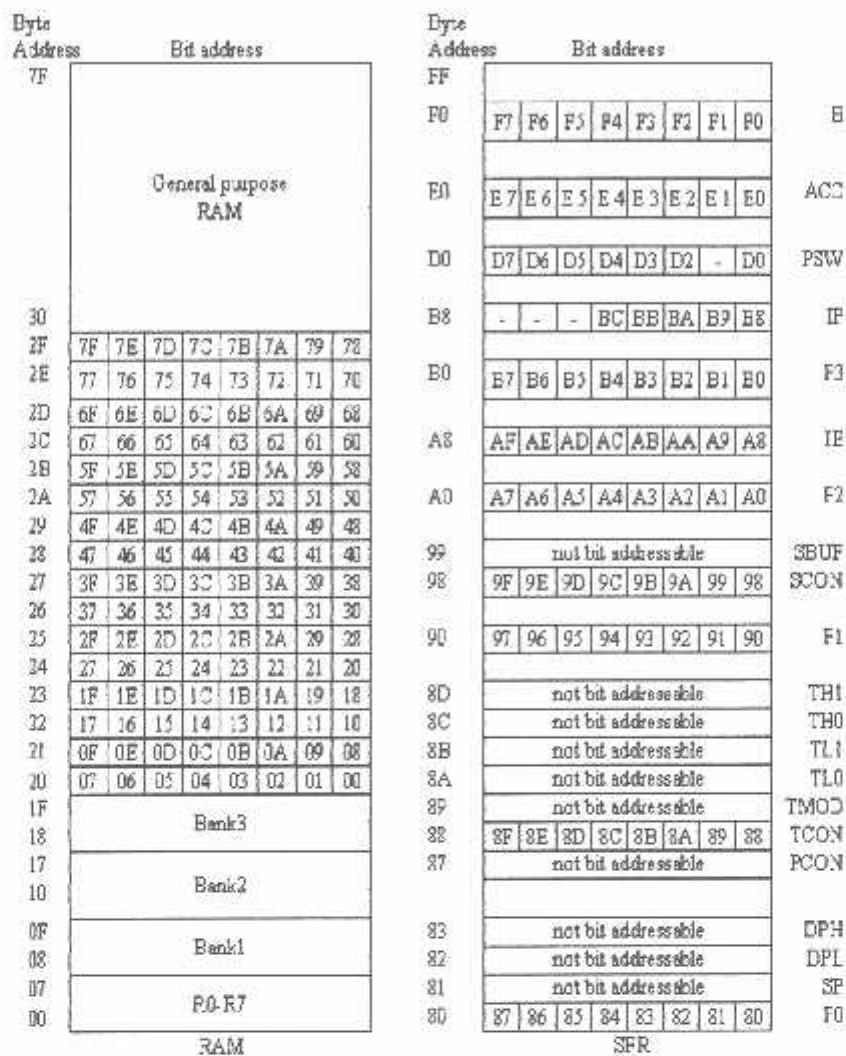
2. Bit Addressable RAM

RAM pada alamat 20H-2FH dapat diakses secara pengalamatan bit (*bit addressable*) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat diset, clear, AND dan OR.

3. *General purpose RAM* (RAM Keperluan Umum)

RAM ini dimulai pada alamat 30H-7FH yang dapat diakses dengan pengalamatan langsung maupun pengalamatan tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang menunjukkan lokasi yang dialamati. Sedangkan pengalamatan secara tak langsung pada lokasi RAM Internal ini yaitu akses data dari memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1. Register R0 dan R1 merupakan dua buah register pada mikrokontroler AT89S51 yang dapat digunakan sebagai pointer dari sebuah lokasi memori pada RAM Internal.

Untuk gambar peta memori RAM dan *Special Function Register* dapat dilihat pada gambar 2.3.



Gambar 2.3 Peta memori dan special function register^[2]

2.2.3. Register Fungsi Khusus (*Special Function Register*)

AT89S51 memiliki 21 *Special Function Register* (Register Fungsi Khusus). Terletak pada alamat 80H-FFH (lihat Gambar 2.3). Beberapa dari register-register ini dapat dialamati dengan pengalamatan bit. Berikut ini adalah register-register yang ada pada *Special Function Register*.

1. Akumulator

Register ini terletak pada alamat E0H, fungsinya yaitu untuk operasi aritmatik dan operasi logika serta untuk pengiriman data ke memori eksternal.

2. Port I/O

AT89S51 memiliki empat buah port, yaitu port 0, port 1, port 2 dan port 3 terletak pada alamat 80H, 90H, A0H dan B0H. Namun jika digunakan kita menggunakan eksternal memori ataupun fungsi-fungsi spesial seperti *external interrupt*, *serial* ataupun *external timer*, port 0, port 2 dan port 3 tidak dapat digunakan sebagai port dengan fungsi umum. Untuk itu disediakan port 1 yang dikhususkan untuk port dengan fungsi umum (Nalwan : 10).

3. PSW (Program Status Word)

Program Status Word atau PSW terletak pada alamat D0H (gambar 2.3).

4. Register B

Dipakai bersama-sama dengan akumulator untuk proses aritmatik, register ini berfungsi juga register biasa, sifatnya *bit addressable*.

5. Stack Pointer

Merupakan sebuah register 8 bit, terletak pada alamat 81H, berisi alamat dari data yang disimpan di-*stack*. *Stack pointer* dapat diedit atau dibiarkan saja mengikuti standar sesudah terjadi reset (Nalwan : 14).

Instruksi-instruksi yang biasa dipakai adalah *push*, *POP*, *Acsll* dan *Lcall*.

6. Data Pointer

Data pointer (DPTR) merupakan register 16 bit dan terletak pada alamat 82H untuk DPL dan 83H untuk DPH. Berfungsi untuk mengakses *source code* ataupun data yang terletak di memori eksternal. Sebagai contoh lihat *listing* program dibawah ini :

```
Mov  A,#01H
Mov  DPTR,#2000H
Mov  @DPTR,A
```

Listing diatas berfungsi untuk menuliskan data 01H kealamat 2000H, langkahnya pertama data diisikan ke akumulator kemudian DPTR yang berfungsi untuk menunjukkan alamat penyimpanan data diisi dengan 2000H. kemudian isi data dari akumulator akan disimpan ke lokasi memori yang ditunjuk oleh DPTR.

7. Register Timer

AT89S51 terdiri dari dua buah 16 bit Timer/counter, yaitu Timer 0 dan Timer 1. Timer terletak di alamat 8AH untuk TL0 dan 8CH untuk TH0 dan Timer 1 terletak di alamat 8BH untuk TL1 dan 8DH untuk TH1.

8. Register Port Serial

Dalam AT89S51 terdapat sebuah *on chip serial port* (Port Serial di dalam keeping), berfungsi untuk berkomunikasi dengan peralatan lain yang menggunakan serial port juga seperti modem, *shift* register dan lain-lain.

Buffer (penyangga) dalam proses komunikasi terletak pada register SBUF di alamat 99H, sedangkan untuk mengatur mode serial dapat dilakukan dengan mengubah isi dari SCON yang terletak pada alamat 98H.

9. Register Interupsi

AT89S51 memiliki lima buah interupsi dengan dua level prioritas interupsi. Interupsi ini selalu nonaktif setiap sistem dalam AT89S51 di reset. Register-register yang berhubungan dengan *interrupti* adalah *interrupt enable register* (IE) atau register pengaktif interupsi terletak pada alamat A8H, berfungsi untuk mengatur aktifnya tiap-tiap *interrupt* dan yang terakhir adalah *interrupt priority register* (IP) atau register prioritas interupsi terletak pada alamat B8H.

10. Register Kontrol Power

Register ini terdiri atas SMOD, berfungsi untuk melipat dua *baud rate* dari port serial, dua buah bit untuk *flag* fungsi umum pada bit ketiga dan bit kedua, *power down* (PD) bit dan *Idle* (IDL) bit. pada *mode idle* hubungan antara CPU (*central processing unit*) dan *internal clock* terputus, tapi port tetap pada kondisi terakhir, ALE dan PSEN menjadi *high*, timer masih tetap bekerja. Mode *idle* berakhir padasaat terjadi interupsi, reset ataupun kondisi-konsisi lain yang me-reset IDL bit.

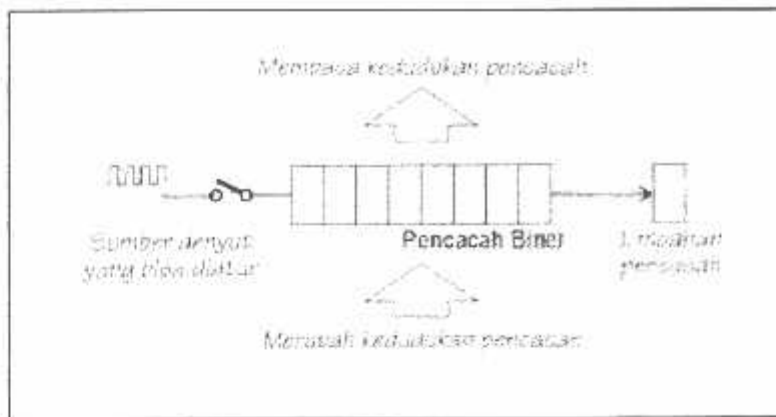
2.2.4 Timer dan Counter dalam MCS51

Timer dan *Counter* merupakan sarana *input* yang kurang dapat perhatian pemakai mikrokontroler, dengan sarana *input* ini mikrokontroler dengan mudah bisa dipakai untuk mengukur lebar pulsa, membangkitkan pulsa dengan lebar yang pasti, dipakai dalam pengendalian tegangan secara PWM (*Pulse Width Modulation*) dan sangat diperlukan untuk aplikasi *remote control* dengan infra merah. Dalam mikrokontroler AT89S51 terdapat dua buah timer yaitu Timer 0 dan Timer 1, keduanya berfungsi sebagai *counter* maupun *timer*.

Pada dasarnya sarana *input* yang satu ini merupakan seperangkat pencacah biner (*binary counter*) yang terhubung langsung ke saluran-data mikrokontroler, sehingga mikrokontroler bisa membaca kedudukan pencacah, bila diperlukan mikrokontroler dapat pula merubah kedudukan pencacah tersebut.

Seperti layaknya pencacah biner, bilamana sinyal *clock* (denyut) yang diumpankan sudah melebihi kapasitas pencacah, maka pada bagian akhir rangkaian pencacah akan timbul sinyal limpahan, sinyal ini merupakan suatu hal yang penting sekali dalam pemakaian pencacah. Terjadinya limpahan pencacah ini dicatat dalam sebuah *flip-flop* tersendiri.

Di samping itu, sinyal *clock* yang diumpankan ke pencacah harus bisa dikendalikan dengan mudah. Hal-hal di atas diringkas dalam Gambar 2.4



Gambar 2.4 Konsep dasar Timer/Counter sebagai sarana input^[3]

Sinyal *clock* yang diumpangkan ke pencacah bisa dibedakan menjadi 2 macam, yang pertama sinyal *clock* dengan frekuensi tetap yang sudah diketahui besarnya dan yang kedua adalah sinyal *clock* dengan frekuensi tidak tetap.

Jika sebuah pencacah bekerja dengan frekuensi tetap yang sudah diketahui besarnya, dikatakan pencacah tersebut bekerja sebagai *timer*, karena keadaan pencacah tersebut setara dengan waktu yang bisa ditentukan dengan pasti.

Jika sebuah pencacah bekerja dengan frekuensi yang tidak tetap, dikatakan pencacah tersebut bekerja sebagai *counter*, keadaan pencacah tersebut hanyalah menyatakan banyaknya pulsa yang sudah diterima pencacah.

Rangkaian pencacah biner yang dipakai, bisa merupakan pencacah maju (*up-counter*) ataupun pencacah mundur (*down-counter*).

Timer/counter sebagai sarana *input* banyak dijumpai dalam mikrokontroler, misalnya mikrokontroler keluarga MCS51, mikrokontroler ini memiliki *timer/counter* di dalam *chip* sebagai sarana *input*.

2.2.5 Komunikasi Data Serial

Untuk komunikasi data serial AT89S51 dilengkapi dengan *on chip serial port* (Port Serial di dalam *keeping*), sistemnya secara *full duplex* sehingga port serial ini masih dapat menerima data pada saat proses pengiriman data terjadi (Nalwan, 39). Saat proses pengiriman ataupun penerimaan data AT89S51 menggunakan SBUF sebagai penyangga (*buffer*), SBUF ini letaknya pada alamat 99H, sehingga jika AT89S51 sedang melakukan proses pembacaan data pertama sedangkan data yang kedua belum diterima secara penuh maka data ini tidak akan hilang.

SBUFF terdiri atas dua buah register yang terletak di alamat yang sama yaitu 99H, kedua register tersebut adalah *transmite register* sifatnya *write only* (hanya dapat ditulis) dan *receive register* sifatnya *read only* (hanya dapat dibaca).

Sistem kerjanya yaitu saat proses penerimaan data dari port serial, data yang masuk ke dalam port serial akan ditampung pada *receive register* dulu, kemudian diteruskan ke jalur *bus* internal pada saat pembacaan register SBUF. Sedangkan saat pengiriman data ke port serial data yang dituliskan dari *bus* internal ditampung pada *transmite register* dulu sebelum dikirim ke port serial.

Komunikasi data serial dalam AT89S51 dibagi menjadi dua, yaitu (1) komunikasi sinkron dan (2) komunikasi data asinkron. Pada komunikasi data serial secara sinkron membutuhkan sinyal clock untuk sinkronisasi data, sinyal clock tersebut akan terpicu pada setiap bit pengiriman data. Sedangkan pada komunikasi data serial secara asinkron tidak memerlukan sinyal clock sebagai sinkronisasi data, tapi memakai *start bit* dan *stop bit* untuk menandai awal dan akhir dari pengiriman/penerimaan data dengan *baud rate* sebagai pemicunya. *Baud rate* ini dibangkitkan oleh bagian pengirim maupun penerima dengan frekuensi yang sama.

2.3. Memori Eksternal

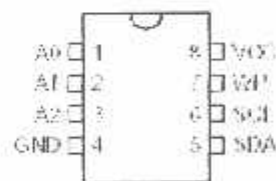
Selain *PEROM* dan *internal RAM* yang terdapat pada 89S51, DT-51 juga mempunyai memori *eksternal* berjenis *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) dengan kapasitas 8 *Kbytes* untuk menyimpan '*user program*' yang di-*download* dari PC atau data. Sesuai dengan namanya maka *EEPROM* dapat ditulis dan dihapus secara elektrik, mirip seperti *RAM* namun bersifat *non volatile* sehingga data yang tersimpan dalam *EEPROM* tidak hilang meskipun catu daya dimatikan.

➤ *EEPROM AT24C08*

EEPROM (*Electrically Erasable Programmable Read Only Memory*) adalah memori yang dapat dihapus dan diprogram ulang. Berbeda dengan *EPROM* biasa, *AT24C08* dapat diperlakukan sebagai *RAM* pada umumnya, namun *EEPROM* jenis ini yang telah ditulis kedalamnya dapat bertahan walaupun catu daya telah dimatikan.

AT24C08 merupakan *chip memory* serial dengan kapasitas 8 *Kbyte* dengan lebar data 8 *bit*.

Selama terjadi siklus tulis, IC ini akan berada pada *state busy* dan tidak akan menjawab sampai suatu data telah selesai ditulis secara lengkap.



Gambar 2.5 *EEPROM AT24C08*^[4]

Fungsi pin-pin dari *EEPROM* AT24C08 dapat dilihat pada tabel 2.3 berikut

Tabel 2.3 Fungsi pim-pin dari *EEPROM* AT24C08 ^[4]

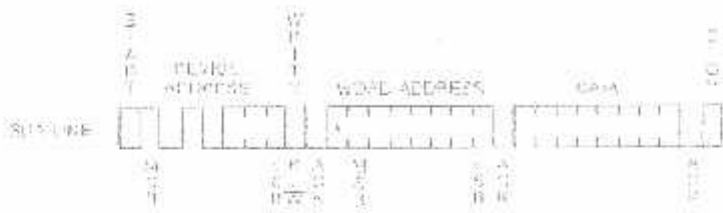
Pin Number	Description
1	A0 - Address Input
2	A1 - Address Input
3	A2 - Address Input
4	GND - Ground
5	SDA - Serial Data
6	SCL - Serial Clock Input
7	WP - Write Protect
8	Vcc - Positive Power Supply

Beberapa pengoperasian dari AT24C08, yaitu:

- a. *Write* : Untuk melakukan penulisan data maka *Pin write* pada *device address* harus diaktifkan terlebih dahulu yaitu dengan pemberian logika (0). Setelah *device addres* diterima, *EEPROM* akan menjawab dengan logika nol dan kemudian menetapkan waktu masuk 8-bit data word yang pertama. Setelah data word diterima *EEPROM* akan menjawab lagi dengan logika nol, baru kemudian data dikirimkan. Bila semua data telah selesai dikirimkan maka harus diakhiri dengan pemberian tanda pemberhentian pengiriman. Untuk lebih jelasnya dapat dilihat pada gambar berikut

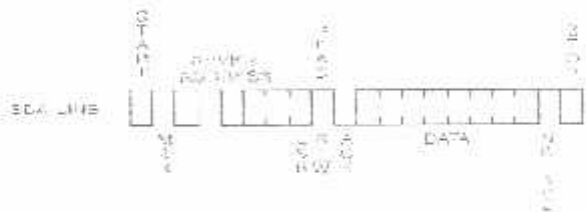


Gambar 2.6 Device Address^[4]



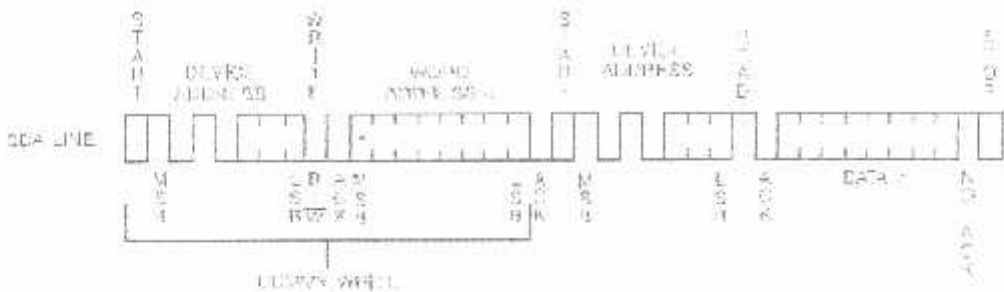
Gambar 2.7 Format Penulisan Data^[4]

b. *Read* : Untuk pengoperasian pembacaan data yang disimpan dimemori maka perlu mengirimkan *device address* untuk pengaktifan pin *Read* yaitu dengan pemberian logika (1).



Gambar 2.8. Format Pembacaan Data^[4]

Untuk pembacaan data secara acak maka, yang harus dikirimkan pertamakali adalah format *write*, tetapi setelah *word address* tidak langsung data melainkan dikirimkan format pembacaan data. Untuk lebih jelasnya dapat dilihat pada gambar 2.9 berikut.

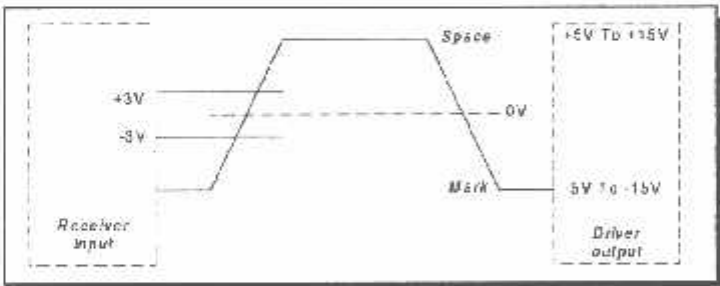


Gambar 2.9. Format Pembacaan Data dengan Alamat Acak^[4]

2.4. Interface Unit RS-232

2.4.1 Karakteristik Elektris

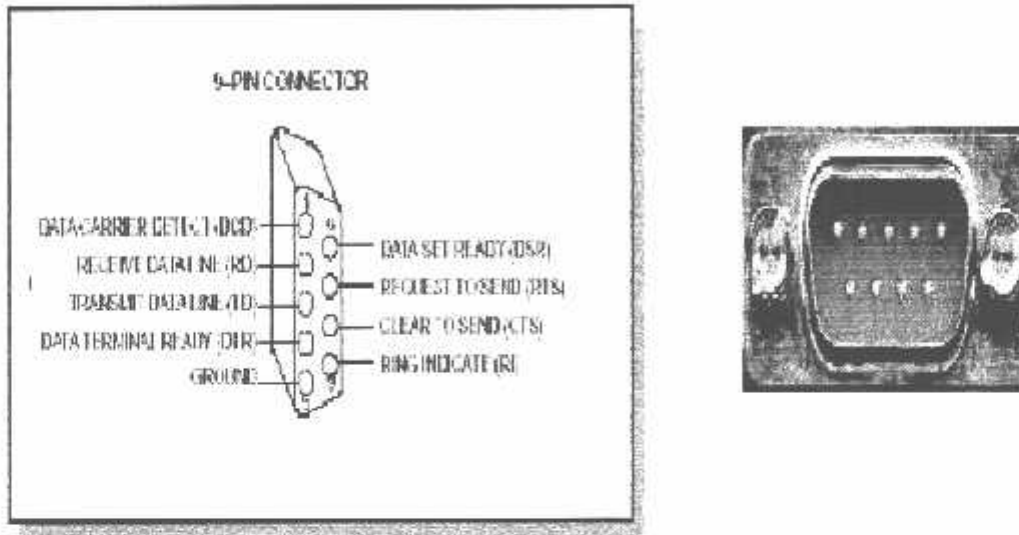
Karakteristik elektris meliputi level tegangan, yaitu mempunyai tegangan keluaran +5 sampai +15 volt untuk mewakili level rendah (logika '0') dan -5 sampai -15 volt untuk mewakili level tinggi (logika '1'). Yang perlu diperhatikan bahwa untuk komunikasi RS232, level tinggi (-3 sampai -15 volt) didefinisikan sebagai logika '1' (*marking*) sebaliknya level rendah (+3 sampai +15 volt) didefinisikan sebagai logika '0' (*spacing*).



Gambar 2.10. Level Logika Standar RS-232^[5]

2.4.2 Karakteristik Mekanis

Untuk kebutuhan *interface*, sinyal RS-232 memanfaatkan sebuah konektor dengan jumlah pin 9 atau 25. Konektor tersebut sering disebut DB 9 *connector*. Gambar tipe konektor DB9 ditunjukkan dalam Gambar 2.11.



Gambar 2.11. Konfigurasi Pin Konektor RS-232^[5]

Fungsi masing-masing pin adalah sebagai berikut :

Tabel 2.4 Fungsi Pin dalam DB9^[6]

No	Nama	Fungsi
1.	DCD (<i>Data Carrier Detect</i>)	Mendeteksi sinyal <i>carrier</i> dari modem lain
2.	RD (<i>Receive Data Line</i>) / (Rx/D)	Pengiriman data serial dari DCE ke DTE
3.	TD (<i>Transmit Data Line</i>) / (Tx/D)	Pengiriman data serial dari DTE ke DCE
4.	DTR (<i>Data Terminal Ready</i>)	Memberitahu DCE bahwa DTE telah aktif dan siap untuk bekerja
5.	<i>Ground</i>	Referensi semua tegangan antarmuka
6.	DSR (<i>Data Set Ready</i>)	Memberitahu DTE bahwa DCE telah aktif dan siap untuk bekerja
7.	RTS (<i>Request To Send</i>)	Memberitahu DCE bahwa DTE akan mengirim data
8.	CTS (<i>Clear To Send</i>)	Memberitahu DTE bahwa DCE siap menerima data
9.	RI (<i>Ring Indikator</i>)	Aktif jika modem menerima sinyal ring pada jalur telepon

Untuk spesifikasi RS-232 dapat dilihat dalam Tabel 2.3

Tabel 2.5 Spesifikasi RS-232^[6]

Keistimewaan	Karakteristik
Jenis operasi	Single ended (tak seimbang)
Jenis penggerak dan Penerima per jalur	1 <i>driver</i> 1 <i>receiver</i>
Data <i>rate</i> maksimum	20 kbps
Panjang saluran maksimum	50 ft (15 m)
Tegangan keluaran penggerak	$\pm 5 - \pm 15$ volt
Sensitivitas penerima	+3 volt

2.5. Komunikasi Serial 485

2.5.1 RS-485

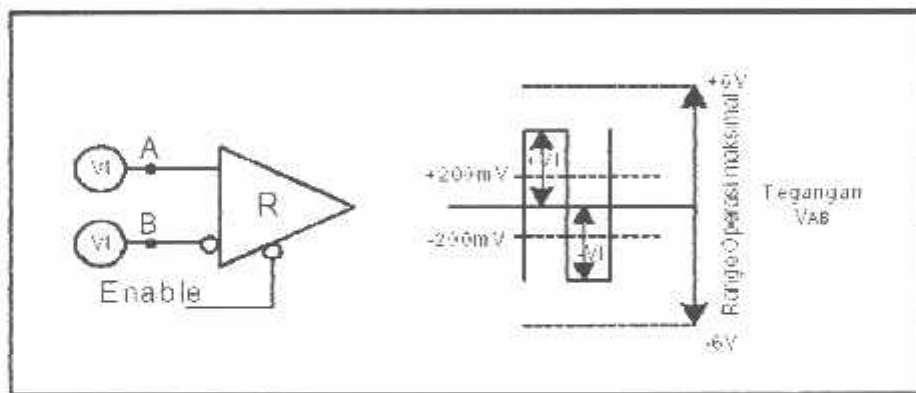
Standar data serial untuk PC dan aplikasi telekom yang paling dikenal adalah RS-232. Mirip dengan RS-232, RS-485 juga standar untuk data serial. Standar tersebut digunakan untuk tujuan kontrol dan instrumentasi yang memerlukan komunikasi. Standar RS-485 dikembangkan oleh dua asosiasi, yaitu: *The Electronics Industries Association* (EIA) dan *The Telecommunication Industry Association* (TIA). EIA memberi label dalam semua standarnya dengan singkatan RS (*Recommended Standard*). Banyak teknisi menggunakannya tetapi secara resmi mengganti RS dengan "EIA/TIA" untuk melihat identitas asal standar tersebut.

Dewasa ini dikenal 2 macam saluran transmisi data, yaitu transmisi saluran tunggal (*single-ended unbalance data transmission*) yang biasa dipakai RS-232, dan transmisi saluran ganda (*differential balanced data transmission*) yang biasa

dipakai RS 485. Dalam saluran jenis saluran tunggal, satu sinyal dikirimkan dengan satu buah kabel ditambah *ground*, atau dua sinyal dikirim dengan dua buah kabel ditambah kabel *ground*. Sedangkan jenis saluran ganda, setiap sinyal dikirim dengan dua buah kabel dan dua sinyal dikirim dengan empat buah kabel. Meskipun *balanced data transmission* lebih rumit, tetapi memiliki sifat yang tahan terhadap gangguan listrik eksternal, sehingga dapat menyalurkan data lebih jauh dengan kecepatan tinggi.

2.5.2 Karakteristik Elektris

Dalam Gambar 2.8 ditunjukkan karakteristik elektrik RS485.

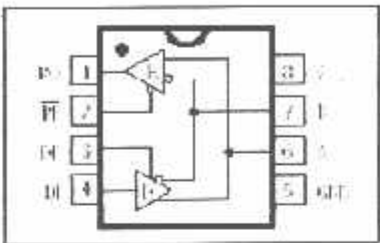


Gambar 2.12. Level Tegangan Masukan Saluran Penerima^[7]

Jika beda tegangan dalam masukan V_{AB} lebih dari $+200\text{ mV}$ (V_A lebih positif daripada V_B), maka port penerima (*receiver*) RS-485 akan berlogika 1. Sebaliknya, jika beda tegangan antara V_{AB} kurang dari -200 mV (V_A lebih negatif daripada V_B), maka receiver akan berlogika 0.

2.5.3 Karakteristik Mekanis

Untuk membantu perancang dalam menggunakan RS-485, maka dibutuhkan suatu komponen untuk memenuhi ketentuan RS-485. Dalam Gambar 2.9 ditunjukkan bentuk komponen IC yang memenuhi RS-485.



Gambar 2.13. Bentuk Mekanis Komponen IC RS-485^[8]

Untuk memulai proses komunikasi half duplex, yang mana pengaturannya melalui DE/\overline{RE} . Jika DE/\overline{RE} diberi masukan logika 1, maka berfungsi sebagai pengirim data (*transmitter*) dan sebaliknya jika diberi logika 0, maka berfungsi sebagai penerima data (*receiver*).

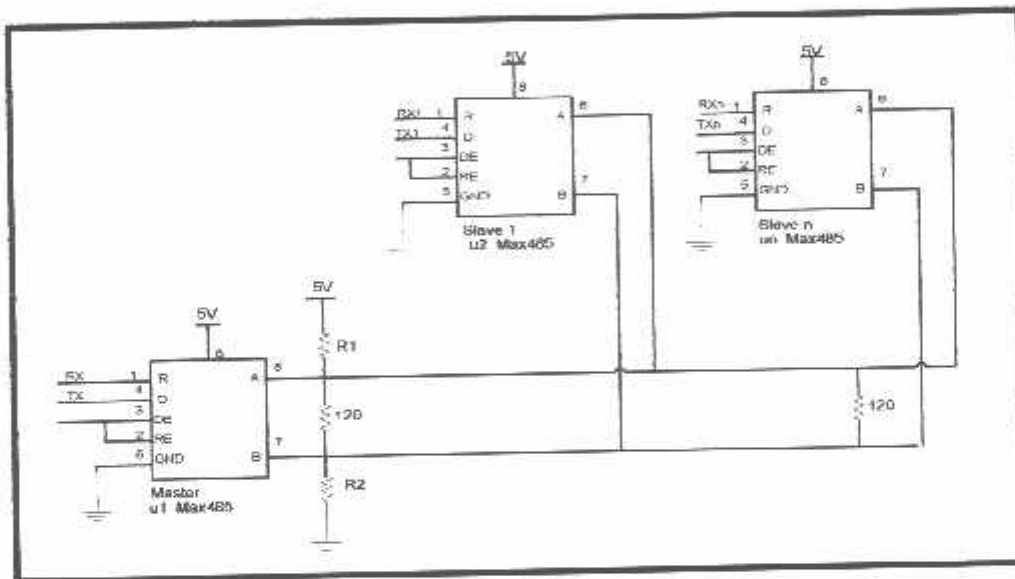
Spesifikasi RS-485 dapat dilihat dalam Tabel 2.4.

Tabel 2.6 Spesifikasi RS-485^[9]

Keistimewaan	Karakteristik
Jenis operasi	<i>Differential (seimbang)</i>
Jenis penggerak dan Penerima per jalur	1 <i>driver</i> 32 <i>receiver</i>
Data rate maksimum	10 Mbps
Panjang saluran maksimum	4000 ft (1200 m)
Tegangan keluaran penggerak	$\pm 1,5$ volt
Sensitivitas penerima	+200 mV

2.5.4 Komunikasi Multi Point Dengan RS-485

Komunikasi *multipoint/ multi drop* dengan RS-485 ada 2 macam, yaitu system dengan 2 *wires* dan system dengan 4 *wires*. Sistem 2 *wires* dapat dijelaskan bahwa hanya diperlukan sebuah IC MAX485 yang dipasang pada masing-masing *master* dan *slave* nya. Terlihat juga pada ujung saluran masing-masing dipasang resistor 120 ohm yang menghubungkan kedua kabel. Resistor tersebut digunakan untuk mengurangi terjadinya gelombang pantul pada saluran yang sering terjadi pada transmisi berkecepatan tinggi.



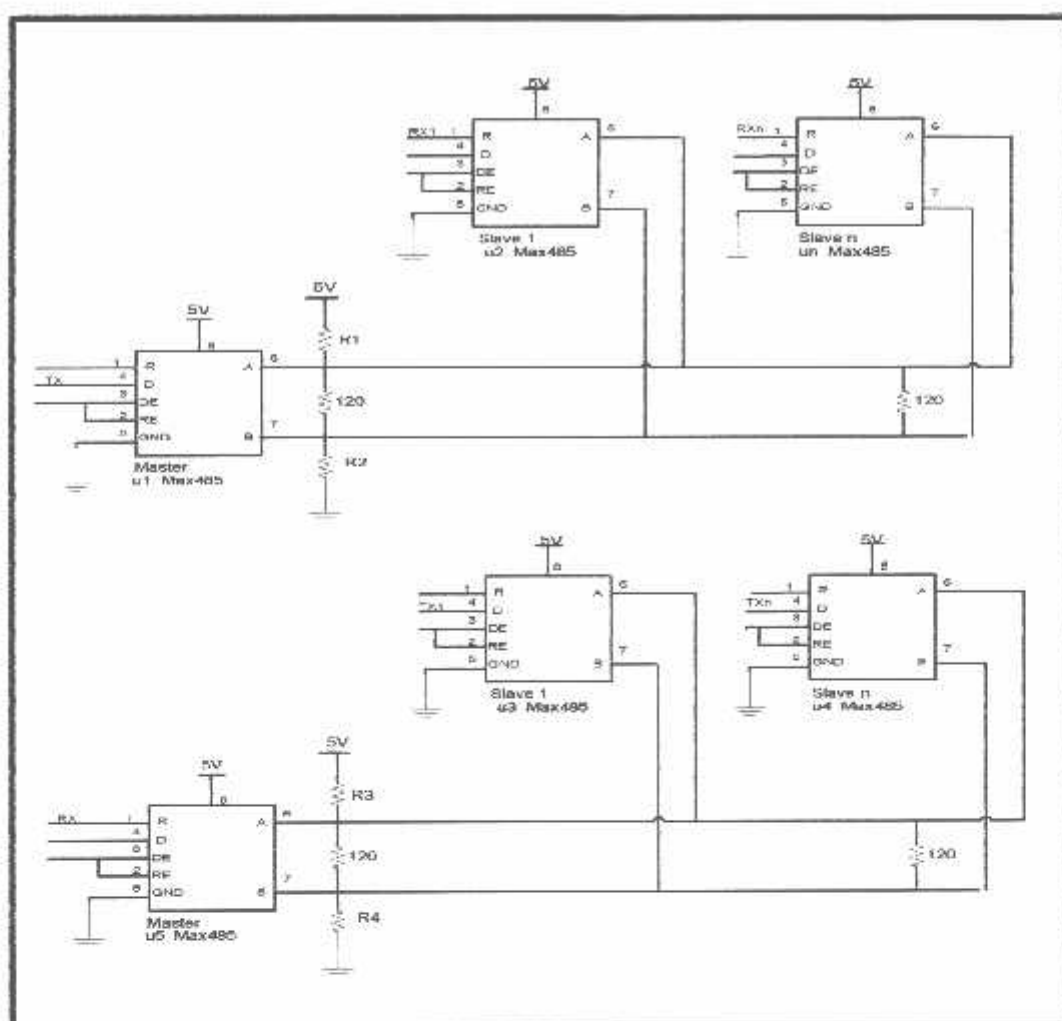
Gambar 2.14 Komunikasi 2 wire^[10]

Pada system 2 *wires*, transmisi data berlangsung *half duplex*, maka dari itu semua pin dari DE dan RE harus dikontrol. Jika hal ini tidak dilakukan akan terjadi kesalahan. Karen transmisi data hanya bias berlangsung secara bergantian. Oleh karena itu *master* lah yang harus mengatur terjadinya proses

komunikasi data. Pada saat *master* akan menghubungi salah satu *slave*, maka *master* harus mengaktifkan *linegenerator* dengan memberi logika *high* pada pin DE dan RE, sementara semua ke *slave* dalam keadaan siap menerima data (RE dan DE masing-masing *slave* diberi logika *low*). Kemudian *slave* yang terpanggil akan membalasnya dengan mengaktifkan *line generator* nya dan mengirimkan informasi ke saluran. Setelah itu *slave* tersebut mematikan kembali *line generator* dan *master* kembali mengaktifkan *line generator* untuk menghubungi *slave* yang lain.

Pada *slave* yang jaraknya paling jauh dari *master* dipasang 2 buah resistor bias yang masing-masing bernilai sama sesuai dengan jumlah *nodenya*. Resistor yang terhubung ke A dihubungkan ke +5V dan resistor yang terhubung ke B dihubungkan ke *ground*. Hal ini digunakan untuk membias arus agar agar tidak terjadi penurunan tegangan akibat beban.

Sedangkan pada system 4 *wires*, dibutuhkan masing-masing 2 buah IC MAX485 pada *master* dan *slave* nya. Keuntungan system 4 *wires* ini adalah proses transmisi data berlangsung secara *full duplex* karena proses transmit dan receive melalui jalur yang berbeda.



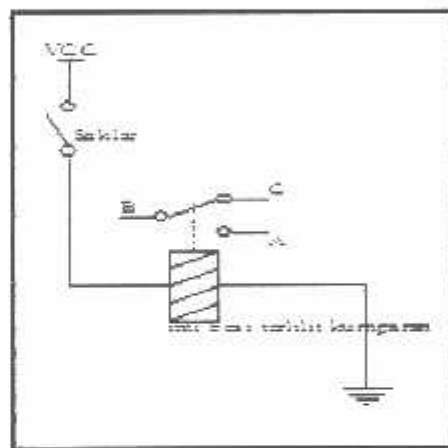
Gambar 2.15 Komunikasi 4 wire⁽¹⁰⁾

2.6. Relay

Relay adalah suatu alat untuk membuka dan menutup kontak secara elektrik dengan tujuan menghubungkan fungsi dari rangkaian satu ke rangkaian lainnya. Relay adalah komponen elektronik yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada sebuah besi lunak.

Cara Kerja :

Jika ada arus yang masuk dan melalui kumparan maka pada kumparan tersebut akan terdapat induksi magnetik. induksi magnetik nantinya akan menarik pegas kontak untuk merubah posisi awalnya menjadi terhubung ke bagian yang diinginkan. Setelah arus berhenti, maka tidak ada induksi sehingga kontak akan kembali ke input semula. Jika kumparan dialiri arus listrik maka inti besi akan menjadi magnet dan menarik pegas sehingga kotak AB terhubung dan BC terputus.

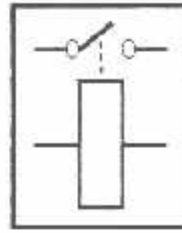


Gambar 2.16 Cara Kerja Relay

Relay merupakan suatu alat untuk menghubungkan atau memerlukan kontak antara komponen yang satu dengan yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang ditimbulkan dari adanya medan magnet listrik yang dihasilkan oleh kumparan yang melilit pada besi lunak.

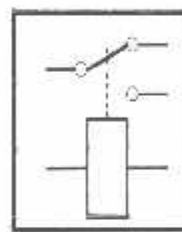
Ada beberapa macam relay, antara lain:

- SPST (*Single Pin Single Terminal*)



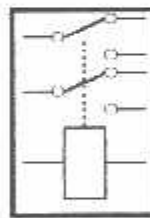
Gambar 2.17 Relay SPST

- SPDT (*Single Pin Dual Terminal*)



Gambar 2.18 Relay SPDT

- DPDT (*Dual Pin Dual Terminal*)



Gambar 2.19 Relay DPDT

2.7. Sensor Cahaya (Lutron LX-1102)

Sensor cahaya adalah alat yang digunakan untuk merubah besaran cahaya menjadi besaran listrik. Prinsip kerja dari sensor cahaya adalah

mengubah energi dari foton menjadi elektron. Idealnya satu foton dapat membangkitkan satu elektron.

Dengan menggunakan sensor cahaya, cahaya yang diterima dapat diketahui berapa intensitasnya. Dari intensitas yang diperoleh dapat diproses untuk pengendali perangkat-perangkat lain sehingga dapat bekerja secara otomatis.

Sensor cahaya (Lutron LX-1102) merupakan alat ukur cahaya yang memiliki kemampuan untuk mengukur besarnya intensitas cahaya dengan range 40, 400, 4000, 40000 dan 400000 Lux. Sensor yang digunakan adalah foto dioda dan *color correction filter*. Untuk tampilan hasil pengukuran menggunakan LCD. Alat ini dapat diseting untuk batas maksimum pengukuran sesuai dengan yang pengguna inginkan (selama masih dalam batas ukur) dan juga memiliki *Auto power off* atau *manual power off*. Serta dapat berkomunikasi secara serial menggunakan RS232.[11]

2.8. Model Referenci TCP/IP

TCP/IP (*Transmission Control Protocol/ Internet Protocol*) bukan merupakan protokol tunggal, dari namanya dapat diketahui bahwa TCP/IP merupakan sepasang protokol. Dengan kata lain, TCP/IP merupakan kumpulan protokol yang bekerja bersama-sama. Perancangan TCP/IP yang asli bermaksud membuat sebuah protokol yang dapat melintasi lingkungan jaringan yang beraneka ragam dan mempunyai kemampuan menjalankan rute ganda untuk sampai ke tujuan akhir. Fleksibilitas merupakan hal yang sangat penting.

Pada mulanya, TCP/IP disebut NCP (*Network Control Protokol*) dan merupakan proyek penelitian dari departemen pertahanan Amerika untuk ARPAnet (*Adveced Research Project Agency Network*). Protokol percobaan tersebut melewati jaringan *packet-switch*. Tujuannya adalah memastikan jika suatu bagian jaringan mengalami kerusakan, transmisi data akan tetap berlangsung melewati rute alternatif lain

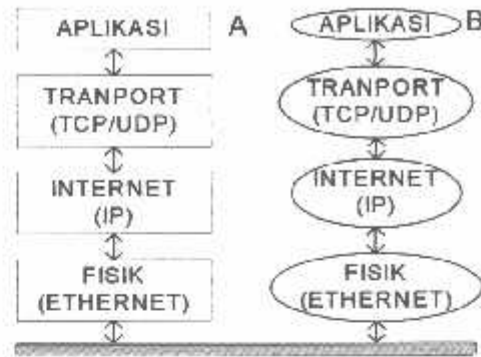
2.8.1. Karakteristik Lapisan TCP/IP

TCP/IP mengatur komonikasi data komputer di internet dan memastikan pengiriman data kealamat yang dituju. Lapisan-lapisan protokol TCP/IP melayani permintaan pengguna untuk mengirim dan menerima data, mengatur komonikasi antar-*host*, melakukan pengecekan kesalahan, menyampaikan paket kealamat yang benar, dan mengirim/menerima data dari media fisik.

2.8.2. Lapisan-Lapisan Model TCP/IP

Protokol model TCP/IP mempunyai empat lapisan yaitu:

1. Lapisan Aplikasi
 2. Lapisan Transport
 3. Lapisan Internet
 4. Lapisan Antarmuka Jaringan/Fisik
-



Gambar 2.20. Lapisan TCP/IP ^[12]

2.8.3.1. Lapisan Aplikasi

Lapisan aplikasi pada model Protokol TCP/IP adalah lapisan yang melayani permintaan pengguna untuk mengirim dan menerima data. Lapisan ini merupakan tempat di mana aplikasi dan servi-servis lain memperoleh akses ke jaringan. Lapisan ini merupakan “jendela” protokol terhadap dunia. Dua API (*Application Programming Interface*/ Antarmuka Program Aplikasi) yang berbeda menyediakan akses ke protokol transportasi TCP/IP, yaitu Socket Window dan Net Bios.

Socket Windows, atau biasa disebut *WinShock*, merupakan jaringan API yang disediakan untuk memfasilitasi komunikasi antara aplikasi TCP/IP dan *stack* protokol yang berbeda. Banyak orang mengembangkan aplikasi untuk digunakan bersama TCP/IP atau sekedar implementasi TCP/IP spesifik milik mereka sendiri. *Socket Windows* diciptakan berdasarkan *socket* API yang pada mulanya dibuat untuk BSD pada sistem operasi UNIX. *Socket Windows* didesain untuk menyediakan landasan dasar untuk aplikasi protokol yang

berada dekat pada bagian atas model referensi TCP/IP. Untuk lebih jelasnya ada pembahasan lebih lanjut mengenai *socket*.

2.8.3.2. Lapisan Transportasi (Transport)

Lapisan transportasi pada model protokol TCP/IP berfungsi mengatur komunikasi antar *host* dan melakukan pengecekan kesalahan. Lapisan ini melakukan dan mempertahankan komunikasi *point-to-point* diantara kedua *host*. Fungsi utamanya adalah memberi balasan terhadap informasi yang diterima, mengontrol aliran, mengurutkan dan mentransmisikan paket-paket data.

Jenis servis yang diperlukan tergantung kepada aplikasi, dapat menggunakan TCP (*Transmission Control Protocol*) atau UDP (*User Datagram Protocol*). TCP biasanya digunakan ketika aplikasi yang dijalankan memerlukan transmisi data dalam jumlah besar dan memastikan data yang diperoleh penerima merupakan data yang benar dan dalam waktu pengiriman yang tepat. Adapun UDP digunakan apabila aplikasi mentransmisikan data dalam jumlah kecil dan tidak memerlukan balasan dari penerima. Pengiriman data melalui UDP bersifat *unreliable* (tidak ada pengecekan kesalahan paket); tidak ada laporan tentang penerimaan data dari penerima.

- ***Transmission Control protocol (TCP)***

TCP bertanggung jawab melakukan transmisi data yang tepat dari suatu *node* pada suatu jaringan ke jaringan lain. TCP membutuhkan suatu sesi koneksi yang terarah atau siklus *virtual* diantara dua komputer, ketika TCP

melakukan koneksi dengan komputer lain, TCP membuat dan mengirimkan paket yang diminta oleh komputer yang menginginkannya. Adapun komputer yang meminta sedang aktif dan mendengarkan, komputer penerima akan merespon paket yang dikirim, dan komputer pertama pun akan merespon. Proses di mana dua komputer melakukan komunikasi seperti ini disebut *three-way handshake*. Disebut *three-way handshake* karena melakukan tiga tahap, yaitu :

1. Klien pertama mengirim *segmen* (paket) berisi nomer *Port* yang akan digunakan dan *Initial Sequence Number* (ISN)-nya kepada *server*.
2. *Server* merespon dengan segmen yang berisi ISN *Server* dan membalas ISN klien ditambah angka satu.
3. Klien menerima ISN *server* yang ditambah angka satu.

Proses *three-way handshake* pertama kali menentukan nomor *Port* yang akan digunakan ISN dari kedua pihak yang berkomunikasi.

Komputer yang melakukan dan mempertahankan koneksi dengan komputer lain harus melakukan pertukaran informasi vital tertentu. Tiap paket TCP berisi nomer *Port* TCP sumber dan tujuan, sejumlah pesan yang dipecah menjadi bagian-bagian kecil, dan pengecekan untuk memastikan bahwa data yang dikirim tidak memiliki kesalahan. Selain itu setiap paket juga berisi nomer tanda terima yang memberi tahukan bahwa potongan-potongan data yang dikirim komputer pengirim telah sampai ditujuan. Tiap paket juga berisi *TCP Sliding Windows metric* untuk mengtur kontrol aliran data antara dua komputer.

▪ *User Datagram Protocol (UDP)*

UDP merupakan lapisan transportasi protokol lain yang bertugas melakukan transmisi data *point-to-point* pada sebuah jaringan. Tidak seperti TCP, UDP merupakan protokol yang bersifat *connectionless*. Tidak ada sesi yang dilaksanakan, protokol ini benar-benar mencoba melakukan usaha terbaiknya ketika melakukan pengiriman data. UDP tidak akan memeriksa apakah *host* tujuan menerima pesan yang sudah dikirim.

UDP digunakan oleh aplikasi yang mengirimkan data dalam jumlah yang kecil, *unreliable* (tidak memerlukan pengecekan kesalahan), dan ketika TCP dianggap terlalu komplek, terlalu lambat, atau tidak dibutuhkan. *NetBIOS name resolution* dan *NetBIOS* bagian *server* paket data merupakan contoh aplikasi yang menggunakan protokol UDP.

UDP juga menggunakan nomor-nomor Port untuk mengidentifikasi suatu proses tertentu dengan alamat IP tertentu. Port UDP berbeda dengan Port TCP sehingga dapat menggunakan nomor-nomor Port yang sama tanpa terganggu oleh servis yang lain.

UDP menyediakan beberapa fungsi yang berbeda dari IP yaitu:

- ❖ Nomor Port. UDP menyediakan 16-bit nomor *Port* untuk melaksanakan proses ganda pada *host* yang sama. Sebuah alamat UDP merupakan kombinasi dari 32 bit *IP address* dan 16 bit nomor Port.
 - ❖ *Checksumming*. Tidak seperti IP, UDP melakukan *checksumming* pada data, untuk memastikan integritasnya. Paket yang gagal dalam *checksumming* akan dibuang. Tidak ada aksi lain yang akan dilakukan.
-

2.8.3.3. Lapisan Internet

Lapisan Internet bertugas menjalankan data di dalam dan diantara jaringan yang berbeda. Router sangat berfungsi pada lapisan protokol ini dan bertugas meneruskan paket data dari suatu jaringan atau *segment* kepada jaringan lainnya.

- **Internet Protokol (IP)**

IP merupakan protokol *connectionless* dan menggunakan paket data untuk mengirimkan data dari satu jaringan ke jaringan lainnya. IP tidak memerlukan ACK (*acknowledgement*) untuk memastikan *host* penerima sudah menerima paket yang dikirimkan. Selama pengiriman paket, *Acknowledgement* berada di sisi kiri atas protokol dan sedang menjalankan proses pada level yang lebih tinggi.

Sebagai contoh, ketika sebuah aplikasi menggunakan UDP (disamping menggunakan TCP), maka menjadi tugas dari *servis* atau aplikasi untuk memastikan data sampai pada tujuan dan dalam urutan yang tepat. Proses transmisi data memang menjadi lebih cepat dan biaya jaringan lebih murah, tetapi memerlukan biaya lebih mahal untuk program aplikasi.

Masing-masing paket IP terdiri dari alamat sumber dan tujuan, *protocol identifier* (yang membantu IP menjalankan paket ke protokol transportasi yang tepat), sebuah *checksum*, dan sebuah TTL (*Time to Live*). TTL adalah satuan yang memberi tahu masing-masing *router* bahwa data yang berada diantara sumber dan tujuan telah mencapai jarak tertentu dalam jaringan. Pada saat data melewati sebuah *router* dan dilanjutkan ke jaringan lain, TTL menjadi

berkurang (satu detik). Definisi lain dari TTL adalah waktu untuk sebuah data berada didalam sebuah antrian sebelum dikirim pada siklus tersebut.

Hal ini mencegah kekeliruan dan kerusakan paket karena berputar di dalam jaringan secara tak batas. Apabila suatu paket tertentu dibiarkan mengembar bebas, *bandwidth* jaringan tentu akhirnya akan turun kelevel yang dapat mencegah tetap berlangsungnya komunikasi.

IP menggunakan proses yang disebut “pertambahan (*ANDing*)” untuk menentukan sebuah alamat IP adalah lokal atau *remote*. Jika IP adalah lokal, IP meminta ARP (*Address Resolution Protocol*) untuk alamat perangkat lunak di komputer tujuan. Alamat ini kemudian digunakan untuk meneruskan informasi secara langsung ketujuannya dari pada menyebarkannya ke jaringan lokal.

Apabila alamat IP telah ditentukan sebagai *remote address*, IP memeriksa tabel siklus lokalnya untuk merotasikannya ke alamat tujuan. Apabila tabel tersebut ada, IP menggunakan rute tersebut untuk mengirim paket. Apabila rute tersebut tidak ada di dalam tabel siklus lokal, IP melanjutkan paket tersebut ke *gateway default* lokal.

Gateway default menguji alamat tujuan tersebut. Jika alamat berada di lokal suatu antarmuka *router*, *router* menggunakan ARP untuk melanjutkan paket data ketujuan akhirnya. Jika *router* memutuskan bahwa alamat tujuan merupakan *remote address* maka TTL akan dikurangi satu, menghitung *checksum* baru dan melanjutkan paket ke *default gateway*-nya. Proses ini berulang terus menerus hingga paket data sampai tujuan dan TTL menjadi nol.

- **Address Resolution Protocol (ARP)**

Address Resolution Protocol adalah lapisan internet pada protokol TCP/IP yang bertugas menentukan alamat komputer yang juga disebut sebagai alamat MAC (*MAC address*). MAC harus cocok dengan alamat IP tertentu. Proses ini disebut sebagai *IP address resolution*.

Sebelum paket IP dilanjutkan ke *host* lain, alamat komputer dari komputer penerima harus diketahui. ARP pertama kali memeriksa *chace*-nya untuk menentukan letak alamat komputer tersebut. Jika alamat tersebut ditemukan, paket dikirim secara langsung ke *host* tujuan menggunakan alamat komputer tersebut. Komputer lain pada jaringan akan melihat paket tersebut, tetapi tidak akan memprosesnya karena alamat mereka tidak ada.

Apabila tidak ditemukan catatan lokasi/peta dalam *chace*, ARP mengirim *ARP request broadcast*. Alamat perangkat keras berada dalam format heksadesimal; alamat perangkat keras untuk *broadcast* adalah FF-FF-FF-FF. Pesan pada *broadcast* di asumsikan seperti ini, “adakah yang menggunakan alamat IP X.W.Y.Z ? Jika ya, kirim alamat perangkat keras anda kepada kami di alamat perangkat keras A-B-C-D. ” Setiap komputer di jaringan lokal tersebut kemudian memproses paket pertanyaan tersebut untuk menentukan apakah dia memakai alamat IP yang sedang dicari Oleh *host* yang sedang mengirimkan *ARP request*. Apabila salah satu komputer di jaringan lokal menggunakan alamat IP tersebut, ia akan membuat dan mengirim paket respon yang menunjukan bahwa ia yang sedang dicari oleh *host* serta memasukan alamat perangkat kerasnya pada data yang dikirimnya. Informasi

ini yang kemudian digunakan oleh *host* pengirim untuk mentransmisikan data selanjutnya kepada penerima.

Jika komputer pengirim menerima balasan, ia akan meletakkan catatan alamat perangkat keras tersebut di *ARP chace* untuk penggunaan dilain waktu dan mengirim paket data. Jika tidak ada respon yang diterima, ARP akan mengulangi seluruh proses untuk memperoleh alamat perangkat keras pada lokal *gateway* dan melanjutkan rotasi paket ke jaringan lain.

- ***Internet Control Message Protocol (ICMP)***

Internet Control Message Protocol digunakan oleh IP dan protokol pada lapisan yang lebih tinggi untuk mengirim dan menerima laporan tentang setatus informasi yang sedang ditransmisikan. ICMP biasanya digunakan di antara dua *router* untuk mengontrol berapa cepat suatu informasi mengalir di antara dua sistem. Jika sebuah *router* mengalami kemacetan lalulintas data dari *host* lain, ICMP akan mengirim pesan adanya kemacetan kepada *router* tempat ia menerima data. Kegagalan ICMP bisa disebabkan oleh sistem yang menerima paket data berada pada kecepatan yang terlalu tinggi melebihi kemampuan prosesnya. Pesan pengontrol ini meminta komputer pengirim untuk mengirim data pada kecepatan yang lebih rendah.

- ***Internet Group Management Protocol (IGMP)***

Host pada jaringan lokal menggunakan *Internet Group Management Protocol* (IGMP) untuk melakukan register pada sebuah grup. *Router* pada suatu jaringan lokal mempertahankan informasi grupnya. *Router* pada sebuah

jaringan menggunakan informasi mengenai grup ini untuk mengirim lapisan-lapisan data ke semua *host* yang terdaftar dalam group tertentu.

Pengiriman data dilakukan secara berlapis-lapis, hampir mirip dengan cara *broadcast* yang merupakan suatu metode pengiriman data kepada banyak *host* secara terus menerus. Teknologi ini digunakan oleh aplikasi seperti *Microsoft NetShow*, suatu program yang digunakan untuk mengirim suara dan gambar ke aplikasi *klien* dan dapat digunakan untuk berbagi macam rapat/pertemuan dan acara melalui internet.

2.8.3.4. Lapisan Antarmuka Jaringan/Fisik

Lapisan antarmuka jaringan merupakan lapisan paling bawah. Lapisan ini bertugas mengirim paket data yang berisi IP. Lapisan ini bekerja dekat dengan ARP untuk menentukan *header* informasi yang tepat yang perlu ditambahkan pada masing-masing *frame*. Lapisan ini membuat *frame* yang cocok dengan tipe jaringan yang digunakan, seperti *Ethernet*, *Token Ring*, atau *ATM*, kemudian meletakkan paket data IP.

2.8.3. Transmisi Data pada Model TCP/IP

Secara singkat transmisi data berlangsung seperti berikut. Setiap protokol yang menerima data dari protokol lapisan di atasnya akan menambah informasi tambahan (*header*) miliknya ke data tersebut. Apabila protokol di lapisan atasnya menerima data dari lapisan lebih bawah maka informasi

tambahan akan berkurang sehingga data yang akan diterima sesuai dengan data aslinya.

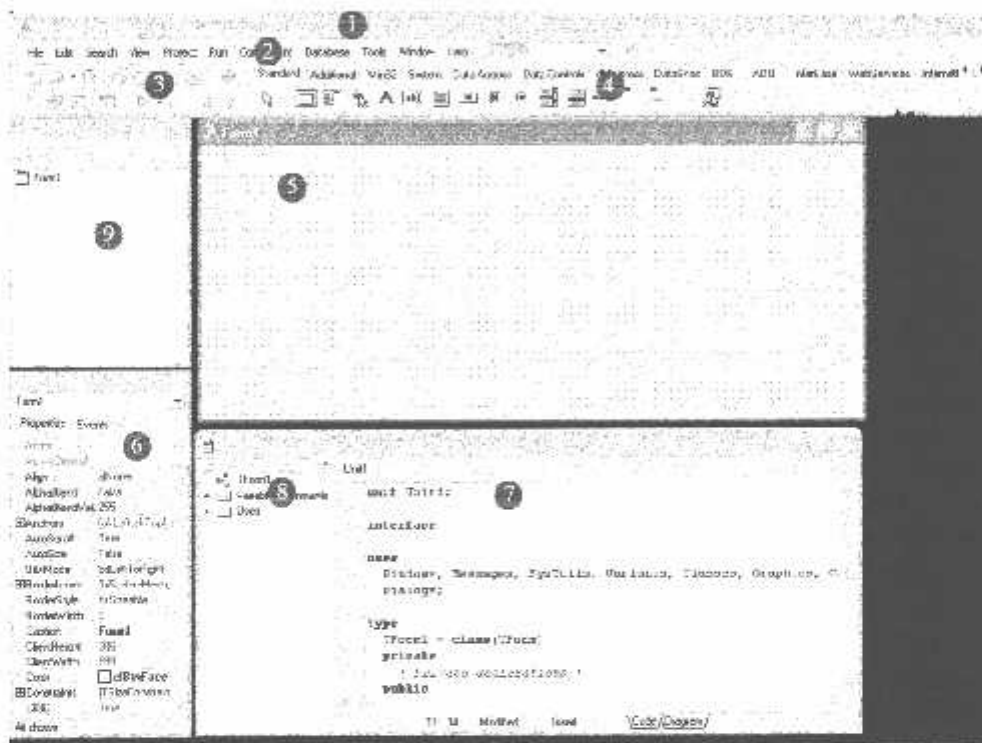
2.9. Borland Delphi 7.0

Delphi adalah salah satu pemrograman visual di lingkungan sistem operasi Microsoft Windows, berbasis bahasa PASCAL, sehingga bila kita telah mengetahui konsep dasar dan aturan-aturan yang berlaku dalam pemrograman PASCAL maka untuk masuk ke Delphi akan lebih mudah. Untuk mempermudah pemrogram dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam dua kelompok, yaitu object dan bahasa pemrograman.

Secara ringkas, object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman Object Pascal. Gabungan dari object dan bahasa pemrograman ini sering disebut sebagai bahasa pemrograman berorientasi object atau *Object Oriented Programming* (OOP).

2.9.1. IDE Delphi

Lingkungan pengembangan terpadu atau *Integrated Development Environment* (IDE) dalam program Delphi terbagi menjadi enam bagian utama, yaitu *Main Window*, *ToolBar*, *Componen Palette*, *Form Designer*, *Code Editor* dan *Object Inspector*. Untuk lebih jelasnya terlihat pada gambar di bawah ini:



Gambar 2.21. Lembar Kerja Delphi^[13]

IDE merupakan sebuah lingkungan dimana semua tombol perintah yang diperlukan untuk mendesain aplikasi, menjalankan dan menguji sebuah aplikasi disajikan dengan baik untuk memudahkan pengembangan program.

Jendela IDE Delphi 7.0 mempunyai perangkat-perangkat yang dapat dipergunakan untuk memudahkan seorang programmer dalam membuat

program. Perangkat-perangkat tersebut seperti terlihat pada gambar 2-10 diantaranya adalah:

1. *Main Window*

Jendela utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, berupa Main Menu, Toolbar dan Component Palette.

2. *Main Menu*

Menu utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi Windows lainnya. Dengan menggunakan fasilitas menu, dapat memanggil atau menyimpan program. Pada dasarnya semua perintah yang diberikan dapat ditemukan pada bagian menu utama ini.

3. *Toolbar*

Dengan *toolbar* dapat melakukan beberapa operasi pada menu utama dengan sebuah klik tunggal. Setiap tombol pada *toolbar* mempunyai sebuah *tooltip* yang berisi informasi mengenai fungsi dari tombol tersebut.

4. *Component Palette*

Component Palette merupakan bagian yang digunakan untuk meletakkan berbagai komponen yang sesuai dengan kategorinya. Pada bidang ini semua komponen yang merupakan bawaan dari Delphi, baik berupa komponen visual maupun komponen nonvisual. Komponen-komponen tersebut berguna untuk mendesain user interface (antarmuka pemakai) dari program yang sedang dibuat. Borland Delphi sendiri memungkinkan

untuk menambahkan komponen dari luar, baik yang dibuat sendiri ataupun komponen dari pihak ketiga (*third party*).

5. *Form Designer*

Merupakan sebuah bidang jendela (*window*) yang masih kosong. Pada bidang ini dapat menempatkan komponen-komponen visual dan nonvisual untuk mendesain *user interface* program. Ketika menjalankan Borland Delphi 7, secara otomatis *Form Designer* akan memanggil sebuah form yang bernama Form1.

6. *Object Inspector*

Dengan perangkat ini dapat mengubah *property* dan *event* pada setiap *object* atau komponen. Object atau komponen yang satu dengan yang lain mempunyai *property* dan *event* yang berbeda. Jika menempatkan sebuah komponen pada suatu Form, komponen tersebut akan berisi nilai *default* dari Delphi. Misalnya komponen TButton akan berisi nilai 25 dan 75 untuk *property* *height* dan *width*. Nilai-nilai *property* tersebut dapat diubah kemudian. Baik pada saat perancangan *interface* program (*design time*) maupun pada saat program berjalan (*run time*) dengan menggunakan kode program.

7. *Code Editor*

Pada bidang ini dapat menuliskan kode-kode program dan logika program dalam bahasa Delphi untuk mengatur jalannya program. Antara Form Designer dan Code Editor merupakan dua bagian yang berkaitan, tidak bias

hanya mendesain user interface pada Form Designer dengan melupakan penulisan kode program pada bagian Code Editor.

8. *Code Explorer*

Pada jendela *Code Explorer* ini akan ditampilkan sebuah *type*, *variabel*, dan *routine* yang didefinisikan pada *unit*. Selain itu juga ditampilkan semua *unit* yang digunakan terletak pada *klausula uses*. Untuk *type* yang kompleks seperti kelas, *code explorer* akan menampilkan semua informasi termasuk daftar *field*, *properties* dan *method*.

9. *Object TreeView*

Merupakan sebuah diagram pohon yang menggambarkan hubungan logis antara komponen visual dan nonvisual yang terletak pada form, data module atau frame. Semua *object* yang anda pakai pada form, data module atau frame akan muncul pada *object TreeView*.

2.9.2. Menu Borland Delphi

1. *Menu File*

Berisi perintah-perintah dasar menu yang sering digunakan yang berhubungan dengan pengoperasian file.

2. *Menu Edit*

Berisi perintah-perintah yang digunakan untuk mentunting teks program dalam jendela code editor, menyunting komponen-komponen yang terletak pada bagian form designer dan beberapa item lainnya.

3. *Menu Search*

berisi perintah-perintah yang digunakan untuk menyunting teks program dalam jendela code editor, menyunting komponen-komponen yang terletak pada bagian form designer dan beberapa item lainnya.

4. *Menu View*

Berisi perintah yang digunakan untuk menampilkan atau menyembunyikan jendela-jendela tertentu dalam program Delphi.

5. *Menu Project*

Berisi perintah-perintah yang berhubungan dengan pengolahan suatu project.

6. *Menu Run*

Digunakan untuk menjalankan program dan melihat jalannya program. Dan juga dapat memantau jalannya program dengan memperhatikan procedure yang dijalankan.

7. *Menu Component*

Digunakan untuk menambah atau menginstall komponen-komponen baru.

8. *Menu Database*

digunakan untuk membuat, mengubah atau melihat database.

9. *Menu Tools dan Help*

Menu Tools digunakan untuk mengubah option atau memanggil Database Dekstop dari menu Delphi.

Menu Help digunakan untuk membuka lembar kerja bantu Delphi.

lainnya. *Socket-socket* Delphi dapat berkoneksi dengan perangkat lain. Komponen tersebut berdasarkan pada TCP/IP dan *socket-socket Windows* tingkat rendah.

Dalam pemrograman internet, dapat menggunakan beberapa alternative berikut :

- Komponen-komponen *socket* Delphi memberikan *interface* yang baik untuk pemakaian langsung *socket-socket* API dari *Windows*.
- Untuk protokol-protokol standar, dapat menggunakan komponen-komponen Indy yang diberikan oleh Delphi.

2.9.4.1. Istilah – istilah

Inti dari internet adalah TCP/IP (*Transmission Control Protocol/Internet Protocol*) yang merupakan kombinasi dari dua *protocol* yang bekerja bersama-sama untuk memberikan *koneksi* pada internet/intranet. IP digunakan untuk mendefinisikan dan mengirimkan *datagram* (*unit transmisi internet*) dan menyatakan skema pengalamatan. TCP bertanggung jawab terhadap servis-servis level atas.

Jika menggunakan jaringan lokal, untuk mencoba program dapat menggunakan IP lokal yang telah diset. Tetapi jika hanya menggunakan sebuah komputer yang akan difungsikan sebagai *server* sekaligus sebagai *client* maka dapat menggunakan alamat 127.0.0.1 (*localhost*) yang merupakan alamat tetap dari komputer. Jika jaringan yang digunakan kompleks, maka harus meminta administrator untuk mengeset alamat yang akan digunakan. *IP address*

merupakan bilangan 32 bit yang dinyatakan dengan empat komponen (*octet*), dan masing-masing dipisahkan oleh sebuah titik. Bilangan tersebut mempunyai arti yang kompleks.

IP *address* tertentu disediakan untuk jaringan internal yang tidak diregistrasi. *Router internet* akan mengabaikan *range* alamat tersebut. Misalnya IP *address* bebas adalah 192.168.0.0 sampai 192.168.0.255, dan dapat dipakai untuk eksperimen pada sebuah jaringan dengan komputer tidak lebih dari 255.

Sebuah IP *address* akan bersesuaian dengan sebuah nama pada *internet*, program *client* akan mencarinya pada sebuah *server domain name*.

Di dalam program bisa menggunakan IP *address* atau nama *hose*. Tetapi nama *hose* akan lebih mudah untuk diingat dan tidak perlu diubah jika IP *address* diubah. Bagaimanapun juga, IP *address* memiliki kelebihan, yaitu tidak perlu proses *mapping*, sedangkan dengan nama *hose* kita harus mencari alamat yang sesuai.

Setiap koneksi TCP dihubungkan ke sebuah port yang dinyatakan dengan bilangan 16 bit. Secara bersama-sama, IP *address* dan TCP port menyatakan koneksi *internet (socket)*. Beberapa proses yang dijalankan pada mesin yang sama tidak dapat memakai *socket* yang sama (port yang sama).

Beberapa TCP *Port* mempunyai pemakaian khusus untuk protokol dan servis tingkat tinggi. Dengan kata lain, harus menggunakan nomor-nomor Port pada saat implementasi servis tertentu. Dalam pembuatan *slave computer* digunakan Port 50 yang berfungsi sebagai *Remote Mail Checking Protocol*. [16]

File *Services* yang berupa file teks mirip dengan file *HOSTS* berisi *port-port* standar yang dipakai oleh *servis-servis*. Pada *servis* dapat ditambahkan data dan pemberi nama. *Socket-socket client* selalu menyatakan nama *port* atau nama *servis* dari *socket server* yang akan dihubungkan.

Protokol adalah kumpulan aturan yang dipakai oleh *client* dan *server* untuk berkomunikasi. Protokol internet tingkat rendah (misalnya TCP/IP) biasanya diimplementasikan oleh *system* operasi. Istilah protokol lebih ditunjukkan untuk *protocol* internet tingkat tinggi seperti HTTP, FTP, SMTP dan lain-lain. Protokol-protokol tersebut didefinisikan didalam dokumen-dokumen standar pada situs *Internet Engineering Task Force* (www.ietf.org).

Jika mengimplementasikan sebuah komunikasi, protokol sederhana dapat didefinisikan (sekumpulan aturan yang menentukan *request client* yang dapat dikirimkan ke *server* dan cara *server* memberi respon pada *request-request* yang berbeda-beda). Protokol-protokol *transver* berada pada *level* yang lebih tinggi dari protokol-protokol *transmisi*.

Cara kerja komunikasi melalui *Socket* adalah program *server* harus dijalankan terlebih dahulu dan menunggu *request* dari *client*. Program *client* akan meminta hubungan ke *server*. Pada saat *client* mengirim *request*, *server* dapat menerima *koneksi* dan memulai *socket* khusus pada sisi *server* yang akan berhubungan dengan *socket* pada *client*.

2.9.4.2. Komponen-Komponen *Socket*

Komponen ini sudah disediakan oleh Delphi tetapi komponen ini tidak ditampilkan pada *Pallete Internet*. Untuk menampilkan komponen ini masuk *Main menu/ Component/ Install Packages* (setelah itu akan muncul *Project Options*)/ pilih *Add* (ambil dari *C:\Program Files/ Borland / Delphi7/ Bin/ dclsockets70.bpl/ Open*)/ *Ok*. Jika langkah ini telah dilakukan, pada *Component Pallete Internet* terdapat komponen *ClientSocket* dan *ServerSocket*. Tampilan dari komponen tersebut dapat dilihat pada gambar berikut:



Gambar 2.22. Tampilan Komponen *ClientSocket*



Gambar 2.23. Tampilan Komponen *ServerSocket*

Untuk penggunaan komponen *socket*, harus memberikan *servis* atau *host* pada sisi *server*, *host* adalah alamat dari komputer, dan pada sisi *ClientSocket* dapat menggunakan dua property yang berbeda untuk menyetingnya yaitu *Host* dan *IP address*. Di dalam *ClientSocket*, *servis* dinyatakan dengan *Port* atau *Srvice*.

2.9.4.3. Koneksi *Socket*

Untuk komunikasi lewat *socket*, pertama-tama harus menjalankan program *server* terlebih dahulu, baru kemudian *client* dijalankan. *Server* hanya menunggu *request* dari *client*, jika *client* mengirim *request* ke *server*, *server* baru memulai *socket server* dan akan menghubungkan ke *socket client*.

Ada tiga jenis koneksi *socket* yaitu:

- **Koneksi *Client***, dimulai oleh *client* dan menghubungkan *socket client* lokal dengan sebuah *socket server remote*. *Socket client* harus menyatakan *server* yang akan dihubungkan dengan memberikan *host* atau alamat *IP* dan *Port*-nya.
- **Koneksi ‘mendengarkan’**, *socket server* bersifat pasif dan menunggu *client*. Jika *client* memberikan *request* baru, *server* akan membuat *socket* baru khusus untuk koneksi tersebut dan ‘mendengarkan’ lagi. *Server socket* ini harus menyatakan *Port* yang menyatakan *sevis* yang diberikan.
- **Koneksi *server***, koneksi yang diaktifkan oleh *server* pada saat *server* menerima *request* dari *client*.

Tipe-tipe koneksi tersebut hanya penting untuk membangun hubungan antara *client* dan *server*. Setelah hubungan terbentuk, kedua sisi bebas untuk membuat *request* dan mengirimkan data.

Jika bekerja dengan *socket* pada *Windows*, *Windows* memberikan beberapa kemungkinan. Pembacaan data dan penulisan data ke *socket* bersifat *asinkron* sehingga proses tersebut tidak memberhentikan *eksekusi* program

BAB III

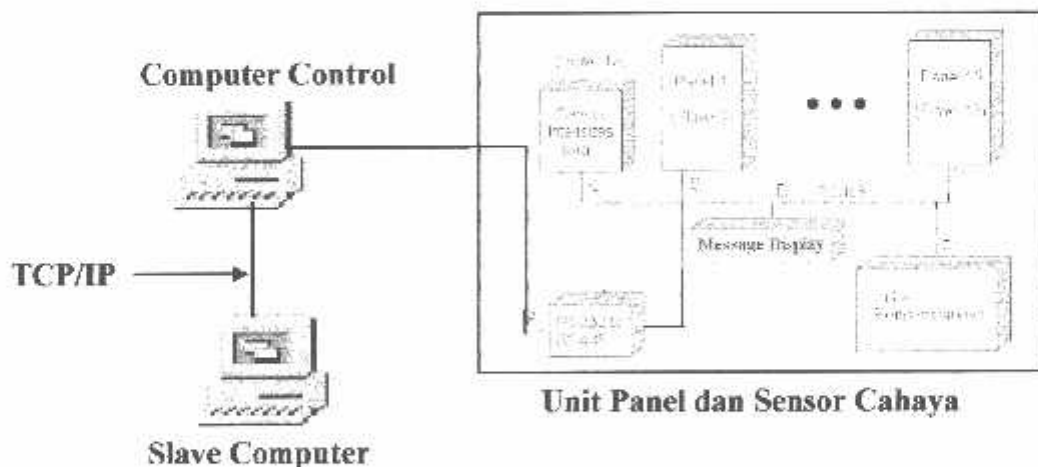
PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan membahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat yang digunakan dalam sistem pengontrolan *computer control* dengan menggunakan *slave* untuk pengoperasian *Smart Building* Jurusan Teknik Elektronika ITN Malang.

3.1.1. Blok Diagram Keseluruhan Sistem

Perancangan dan pembuatan alat ditunjukkan dengan gambar blok diagram dibawah ini :



Gambar 3-1. Diagram Blok Keseluruhan Sistem

Keterangan fungsi dari masing-masing blok diagram diatas sebagai berikut :

- ***Computer Control***

Berfungsi sebagai *master control* untuk pengendali Panel, yang ada pada *smart building* Jurusan Teknik Elektro ITN Malang, disamping sebagai monitoring status tiap panel, juga berfungsi untuk mengontrol peralatan yang terpasang pada jaringan, jadi *Computer Control* dapat mengubah status yang terjadi pada tiap-tiap Panel.

Semua alat yang terpasang pada jaringan (selain *Slave Computer*) tidak boleh mengirim data ke *Computer Control* kecuali diminta oleh *Computer Control* dengan cara *Computer Control* menghubungi alamat Panel terlebih dulu.

Computer Control dalam dalam pengoperasia *smart buiding* dapat dikontrol dan dimonitor menggunakan *Slave Computer*. *Computer Control* tidak akan memberikan data status tiap panel dan sensor kepada *Slave Computer*, jika *Slave Computer* tidak meminta

- ***Slave Computer***

Slave Computer adalah sebuah komputer yang digunakan untuk memonitor dan mengontrol kerja dari *Computer Control* dalam pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang.

- ***Unit Panel dan Sensor***

adalah unit-unit Panel dan sensor cahaya yang terdapat pada *Smart Building* Jurusan Teknik Elektro ITN Malang.

Setiap unit Panel selain dapat bekerja dengan dikendalikan oleh *Computer Control*, juga dapat bekerja sendiri tanpa *Computer Control*.

Jika jaringan listrik padam, maka semua Panel akan menyimpan data terakhir, kecuali peralatan tertentu seperti sensor intensitas sinar.

- **RS-485**

Yaitu komunikasi data yang digunakan antara *Computer Control* dengan unit panel dan sensor cahaya yang ada pada *Smart Building* Jurusan Teknik Elektro ITN Malang.

- **TCP/IP**

komunikasi antara *Computer Control* dengan *Slave Computer* menggunakan Jaringan TCP/IP.

3.2. Prinsip Kerja Alat

Kerja dari *slave computer* adalah untuk mengontrol dan memonitor kerja dari *Computer Control* dalam pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang. *Slave Computer* dapat berkomunikasi dengan *Computer Control* menggunakan jaringan TCP/IP. Dalam perancangan ini media transmisi yang digunakan adalah kabel TP (*Twister Pair*) dengan jenis UTP (*Unshilded Twister Pair*)

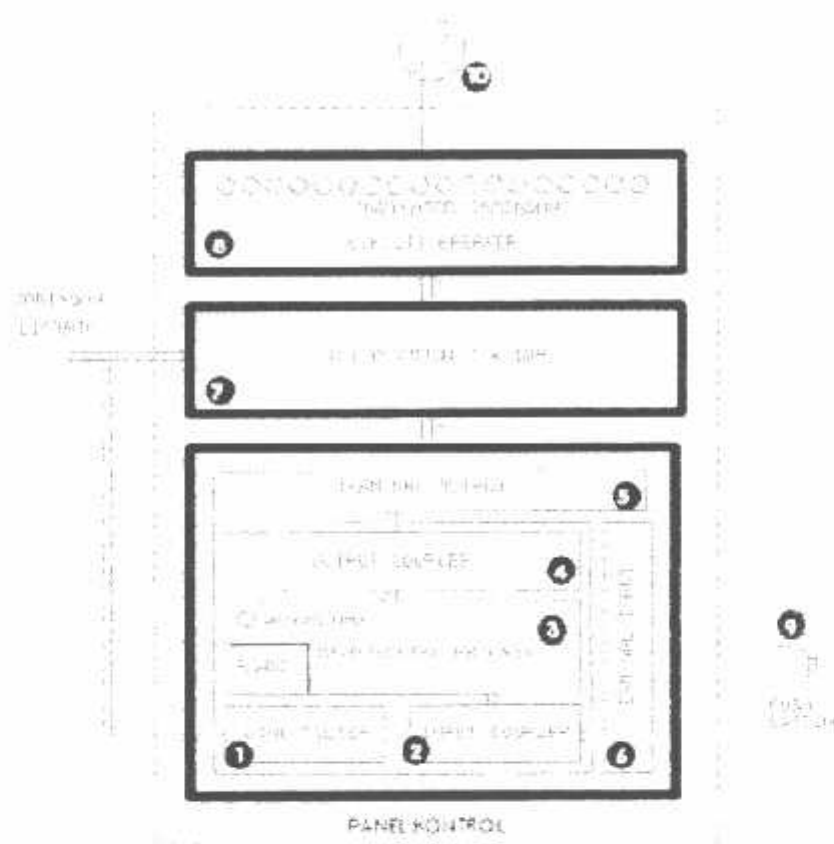
Slave Computer dapat bekerja pada saat *Computer Control* dalam keadaan aktif. Pada saat program dijalankan *Slave Computer* akan meminta kepada *Computer Control* data status dari tiap panel. Jika *Slave Computer* menginginkan perubahan status Panel, *Slave Computer* akan mengirimkan data pada *Computer*

Control untuk melakukan perubahan status, kemudian *Computer Control* akan mengirimkan data pada Panel untuk melakukan perubahan status. Panel akan memberikan jawaban kepada *Computer Control*. Data setiap Panel yang diterima *Computer Control* baru akan dikirimkan pada *Slave Computer* jika diminta.

3.3. Perangkat Keras (Unit Panel dan Sensor yang ada pada *Smart Building* Jurusan Teknik Elektro ITN Malang).

3.3.1. Blok Diagram Unit Switching Panel

Switching Panel merupakan unit yang berfungsi untuk mengontrol beban, baik berupa beban lampu maupun beban lainnya. Unit ini dikontrol oleh mikrokontroler AT89S51 dan dilengkapi dengan jalur serial komunikasi untuk komunikasi dengan *Computer Control*. Gambar diagram blok Panel kontrol ditunjukkan pada gambar 3.2 dan Gambar diagram blok unit switching panel ditunjukkan pada gambar 3.3



Gambar 3.2. Blok Diagram Panel Kontrol^[7]

Keterangan tiap Blok:

1. Line Filter

Blok ini sangat penting karena berfungsi mencegah interferensi (EMI) yang datang dari jaringan input listrik, sekaligus untuk mengisolasi dengan jaringan utama.

2. Input Coupler

Berguna untuk mencegah EMI yang diterima oleh transmisi perkabelan input yang cukup panjang.

3. Mikrokontroler Unit

Adalah unit utama yang mengontrol seluruh sistem. Unit ini juga yang melakukan komunikasi dengan *Computer Control* melalui transmisi data RS485. Didalam unit ini juga terdapat watch dog timer yang bisa menghindari kemacetan (hang up) karena sebab-sebab yang tidak diketahui, sistem mampu mereset dirinya sendiri dengan tetap menjaga kondisi terakhir termasuk juga bila listrik tiba-tiba padam dan kemudian menyala kembali tetap mempertahankan kondisi terakhir. Data kondisi terakhir disimpan pada serial EEPROM.

4. Output Coupler

Adalah bagian yang berfungsi untuk mengisolasi unit mikrokontroler dengan unit keluaran agar tidak terjadi kerusakan, bila unit keluaran mengalami gangguan.

5. Terminal output

Adalah titik-titik sambungan untuk diteruskan ke bagian Relay. (bagian yang berfungsi untuk menghubungkan jaringan listrik ke beban dan setiap keluaran dilengkapi dengan MCB untuk mengamankan jaringan bila terjadi short circuit).

6. Terminal input

Adalah titik-titik sambungan untuk dihubungkan ke saklar / push switch.

7. Unit Relay (Electronic Switch)

Adalah unit switching yang menghubungkan jaringan listrik ke beban lampu.

8. *Circuit breaker dan indikator jaringan*

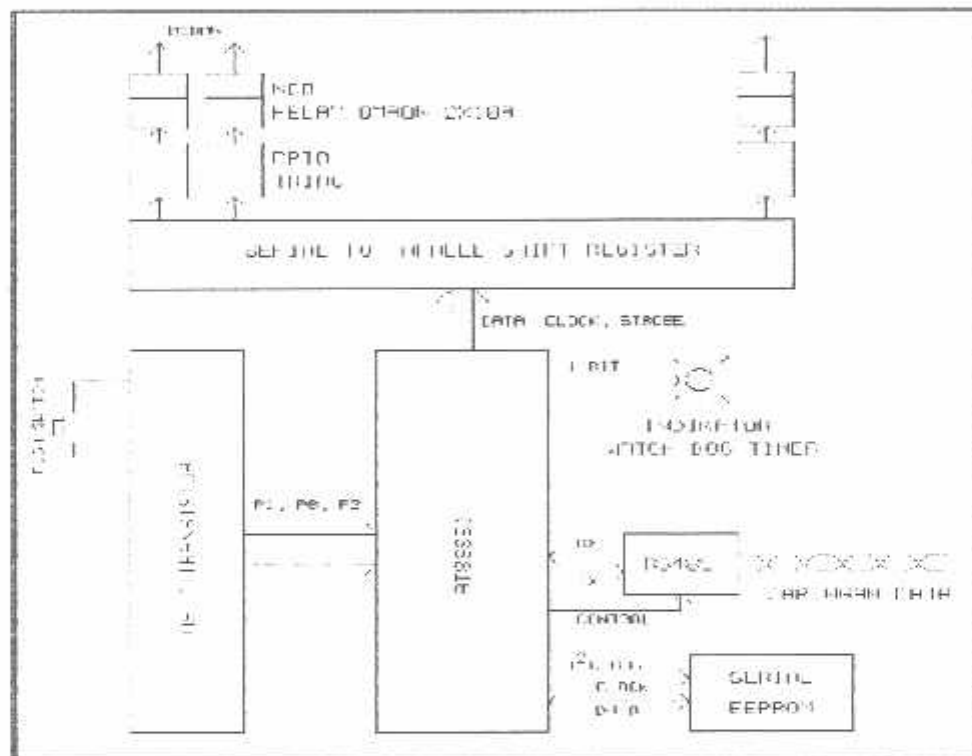
Adalah bagian yang berfungsi mengamankan jaringan bila ada short circuit atau gangguan pada jaringan beban.

9. *Push switch*

Adalah saklar on/off dari lampu pada ruangan tersebut.

10. *Beban lampu*

Adalah lampu pada ruangan dan titik sambungan tertentu.

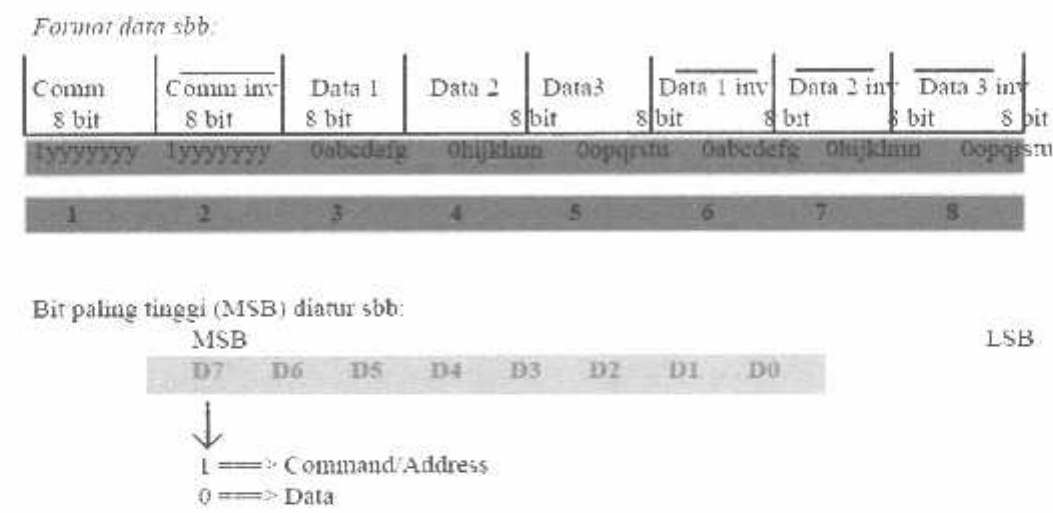


Gambar 3.3 Blok diagram unit switching panel^[17]

3.3.2. *Komunikasi Data antara Computer control dan Panel*

Komunikasi data antara *Computer Control* dengan unit *Panel* dan sensor cahaya dilakukan dengan komunikasi data serial *standard RS 485*. Dengan

pertimbangan posisi *Computer Control* dan *Panel* jaraknya sangat berjauhan. Adapun format yang digunakan dalam komunikasi, mengikuti format seperti pada gambar 3.4.



Gambar 3.4 Format Data Komunikasi Serial^[17]

Format data :

- 1. Lebar data : 8 bit (1 Byte)
- 2..Panjang frame data : Flexible (min 2 x 1 byte)
- 3. Baud rate : 4800 bps
- 4. Parity : none
- 5. Stop bit : 1
- 6. Transmisi data : RS485

Dimana data byte ke :

- 1 – Command atau Address (tiap panel mempunyai address yang berbeda).

- 2 = Data inverse dari byte ke 1 kecuali bit MSB.
- 3 = Data informasi pertama status lampu tiap-tiap ruang
- 4 = Data informasi kedua status lampu tiap-tiap ruang
- 5 = Data informasi ketiga status lampu tiap-tiap ruang
- 6 = Data inverse dari byte ke 3 kecuali bit MSB
- 7 = Data inverse dari byte ke 4 kecuali bit MSB
- 8 = Data inverse dari byte ke 5 kecuali bit MSB

a s/d u adalah status lampu tiap-tiap ruang pada panel tertentu y adalah ommand/address tiap-tiap panel.

Status Lampu a s/d u \implies 0 = padam / OFF
1 = hidup / ON

Tabel 3.1 Kode Panel^[17]

	COMMAND	COMMAND INVERSE		PANEL	KETERANGAN
1	1100 0001	1010 1110	RW	1	Baca status
2	1000 0001	1110 1110	WR	1	tulis status
3	1100 0010	1011 1101	RW	2	Baca status

➤ Untuk baca status, *Computer Control* hanya mengirim *command* dan *command inverse* saja.

- Untuk tulis status, *Computer Control* mengirim frame data lengkap sebanyak 8 byte.
- Untuk menjawab atas pertanyaan *Computer Control*, *Panel* mengirim ke *Computer Control* berupa Data 1 s/d Data 3 inverse.

Setiap pengiriman data baik *Computer Control* maupun *Panel* wajib menguji tingkat kebenaran dari data yang diterima dengan cara membandingkan data *inverse* yang dikirimkannya. Bila terjadi kesalahan, data tidak perlu diproses, harap menunggu kembali sampai menerima *frame* baru yang dikirim berikutnya.

Computer Control harus memberi time out selama kurang lebih 500 ms bila *Panel* yang dihubungi tidak menjawab, dan beri peringatan 'Fail/Rusak utk *Panel* no. #' di layar monitor bila ternyata sampai 5 kali pertanyaan tetap tidak menjawab. Demikian juga *Panel* harus mempunyai *time out* bila setelah *command* pertama diterima namun data yang lain terputus atau rusak/salah, sehingga *Panel* tidak macet pada *routine* tertentu karena menunggu *Computer Control*, *time out* dibuat lebih lama yaitu sekitar 2 detik.

3.3.3. Sensor Cahaya (LX-1102)

Alat ukur ini hanya bekerja atas perintah dari *Computer Control*, bila tidak dipakai maka alat dibuat OFF dengan tujuan untuk efisiensi battery. Langkah pertama adalah *Computer Control* memerintahkan agar alat ukur intensitas cahaya mengambil data, yaitu dengan mengirimkan kode baca intensitas atau "1100 1110 1011 0001" dan bila alat ukur ini pertama kali dihubungi atau sudah lebih dari 10 menit tidak dihubungi maka dia akan mengirimkan jawaban :

" 00000000 00000000 00000000 00000000 00000000 11111111 11111111 11111111 11111111 11111111"

Jadi data akan tersedia setelah 25 detik alat dikunjungi dan sebelum 10 menit setelah kunjungan terakhir.[17]

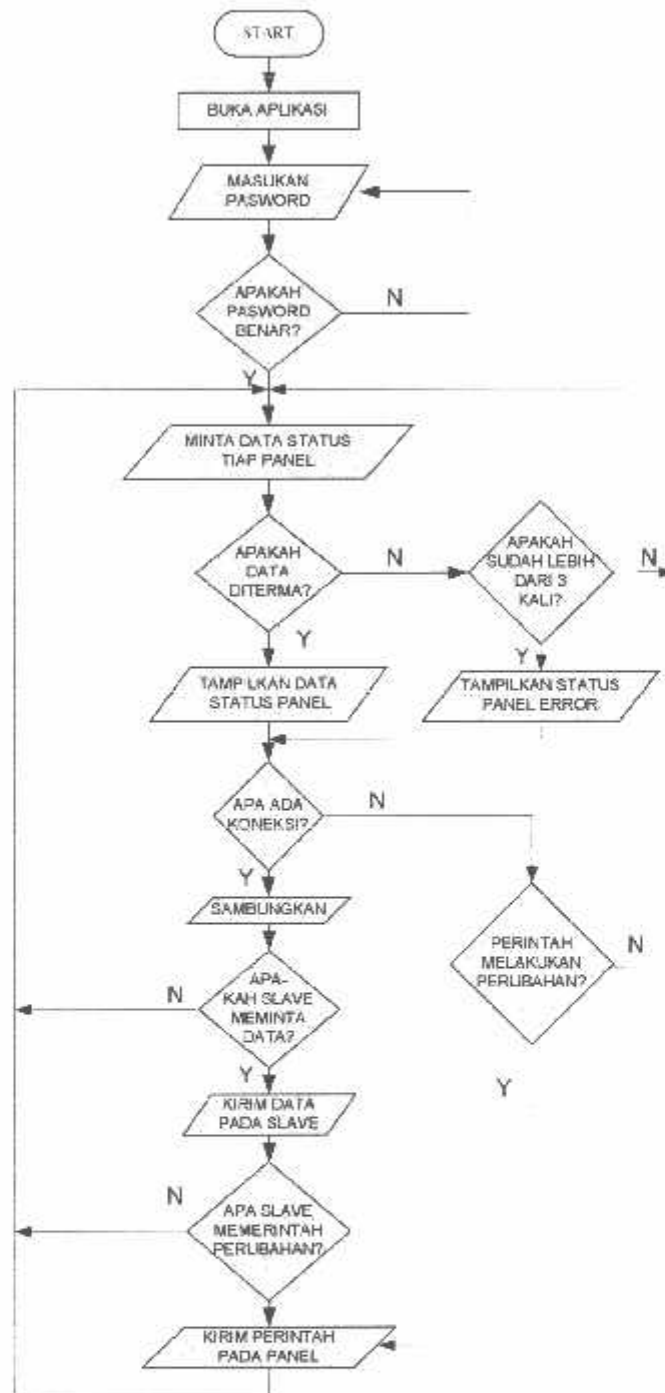
3.4. Perancangan Perangkat Lunak (*Software*)

Perangkat lunak adalah sebuah jembatan yang menghubungkan keseluruhan komponen yang ada pada sebuah komputer. Pada perancangan ini hanya membahas perancangan perangkat lunak untuk komputer menggunakan *Delphi7*.

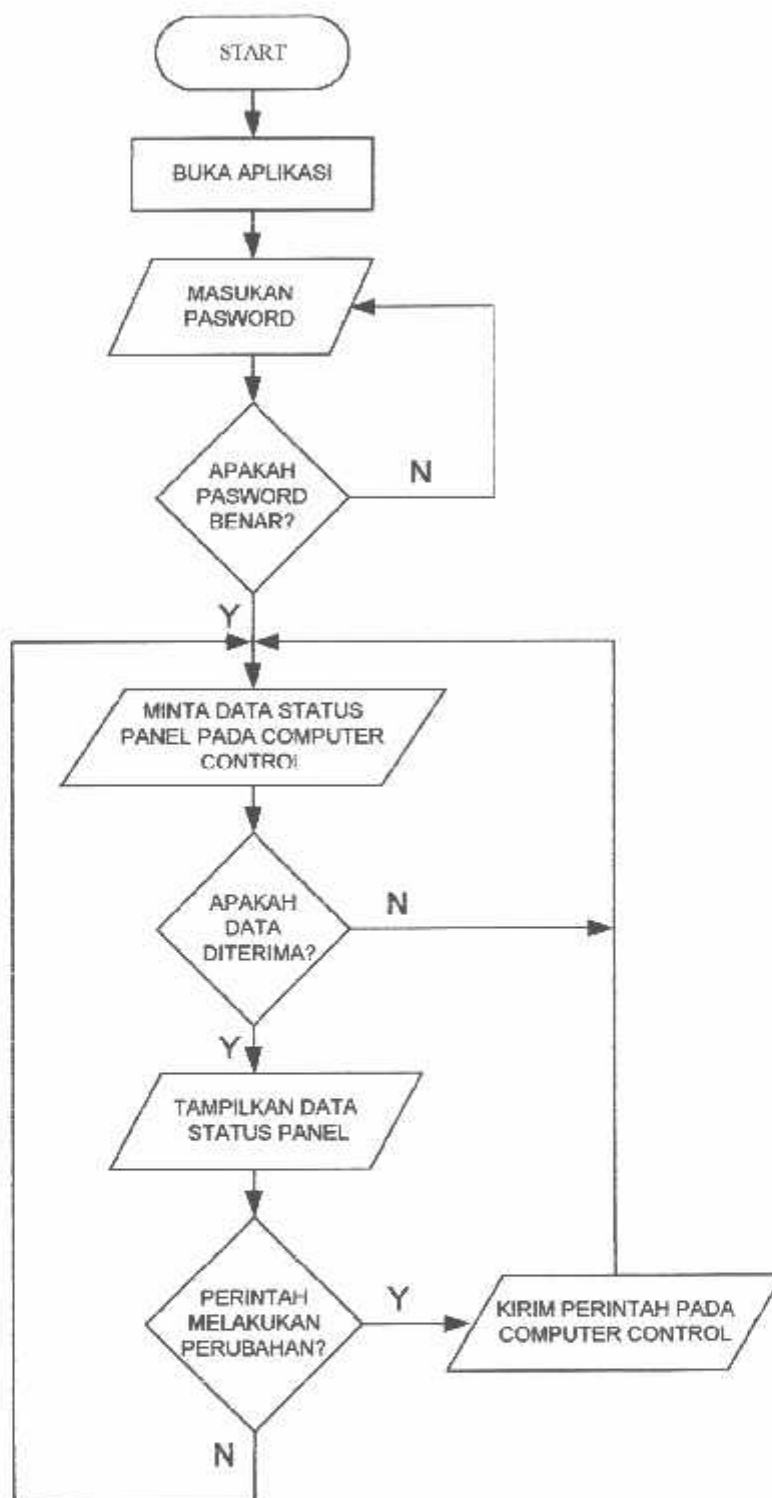
3.4.1. Program Aplikasi Komputer

Program aplikasi adalah *software* di dalam komputer yang berfungsi untuk melakukan pengendalian. Program komputer ini bertujuan untuk mengorganisasi komunikasi antara *Computer Control* dan *Slave Computer*, sehingga perangkat lunak dalam program komputer perlu diketahui terlebih dahulu aturan-aturan atau protokol komunikasi yang digunakan untuk mengatur jalanya komunikasi antara *Computer Control* dan *Slave Computer* dengan Unit Panel dan Sensor. Dalam perencanaan ini menggunakan bahasa pemrograman *visual* yaitu *Delphi7*. Agar komponen-komponen *Delphi7* yang digunakan untuk penunjang perancangan sistem alat dapat bekerja dengan baik, maka ada beberapa komponen tambahan yang diperlukan untuk digunakan dalam perancangan sistem alat ini.

3.4.2. FlowChart



Gambar 3.5. Diagram Alir Program *Computer Controll*



Gambar 3.6. Diagram Alir Program Slave Komputer

3.4.3. Perancangan Pembuatan Software Menggunakan Bahasa Pemrograman *Borland Delphi7*.

Dalam perancangan Pembuatan *Software* untuk *Slave Computer* diperlukan susunan desain komponen-komponen yang akan digunakan sehingga dalam pengoperasiannya nanti akan dapat digunakan dengan lebih optimal. Oleh sebab itu desain dari tampilan program juga sangat diperlukan.

3.4.3.1. Penyusunan Desain *Software*

Dalam penyusunan desain tampilan (dalam 1 form) diperlukan beberapa pembagian halaman agar nantinya dapat mempermudah pengguna dalam pengoperasian. Dalam pendesainan *Software* untuk *Slave Computer* ini dalam 1 form dibagi menjadi enam bagian yaitu:

1. Halaman untuk pengoneksi dengan *Computer Control*
2. Halaman untuk Penyetingan
3. Halaman untuk lantai satu
4. Halaman untuk lantai dua
5. Halaman untuk lantai tiga
6. Halaman untuk lantai empat

Untuk pembagian Halaman dalam satu form digunakan komponen *PageControl*. Untuk membagi, setelah mengambil komponen *PageControl* kemudian meng-klik kanan pada *mouse* dan memilih *new page* sebanyak enam kali sehingga akan terbentuk enam halaman *TabSheet*, sesuai dengan perencanaan.

1. Menyusun Tampilan Desain Koneksi dengan *Computer Control*

Menyusun desain koneksi seperti gambar berikut:



Gambar 3.7. Menyusun Tampilan Form Desain untuk Koneksi dengan *Computer Control*

Untuk mendesain tampilan seperti di atas membutuhkan satu halaman *TabSheet*, satu buah komponen *Panel*, satu buah komponen *RadioGroup* dan tiga buah komponen *Label*. Kemudian merubah properti dari masing-masing komponen seperti tabel dibawah.

Tabel 3.2 Merubah isi Properti pada Layar Connect ke Control

Komponen	Properti	Isi
<i>PageControl (TabSheet)</i> (Pada palet Win32)	Name	TabSheet6
	Caption	Connect Ke Com.Control
Panel (pada palet standar)	Name	Panel4
	Caption	(kosong)

<i>RadioGroup</i> (pada palet standar)	Name	Client
	Caption	Koneksi
	Items	Aktifkan; dan Putuskan.
<i>Label</i> (pada palet standar)	Name	Label32
	Caption	S t a t u s :
<i>Label</i> (pada palet standar)	Name	Label29
	Caption	Status ComputerControl :
<i>Label</i> (pada palet standar)	Name	Label30
	Caption	Terputus
	Color	Red
<i>ClientSocket</i> (pada palet Internet)	Name	<i>ClientSocket1</i>
	Address	127.0.0.1 (alamat tetap Computer)
	Port	50
	Active	False
<i>ServerSocket</i> (pada palet Internet)	Name	<i>ServerSocket1</i>
	Port	50
	Active	True

Memasukan Program berikut kedalam *RadioGroup(name : client)* pada **Event OnClick**

```
procedure TForm_Utama.ClientClick(Sender: TObject);
begin
  IF Client.ItemIndex = 0
  Then
    Begin
      ClientSocket1.Address := InputBox('Setup Koneksi','IP Server','');
      ClientSocket1.Active := True;
      Label30.Caption := 'Terhubung';
      Label30.Font.Color := clGreen;
    End
  Else
    Begin
      ClientSocket1.Active := False;
      Label30.Caption := 'Terputus';
```

```

    Label30.Font.Color := clRed;
  End;
end;

```

Penjelasan :

OnClick artinya Jika *Radiogroup (Client.ItemIndex)* diklik maka program yang ada didalamnya akan dijalankan. Yaitu bila *Client.ItemIndex = 0*, maka akan muncul *InputBox* untuk memasukan alamat IP dari *Computer Control* yang akan dikoneksikan (disambungkan). Tetapi jika *Client.ItemIndex* tidak sama dengan nol maka Koneksi *Socket* akan diputuskan.

Memasukan Program berikut kedalam **Event *ServerSocket1ClientRead***

```

procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
Var x: string;
Begin
  x := Socket.ReceiveText;
  Cek_Terima(x);
  x := "";
end;

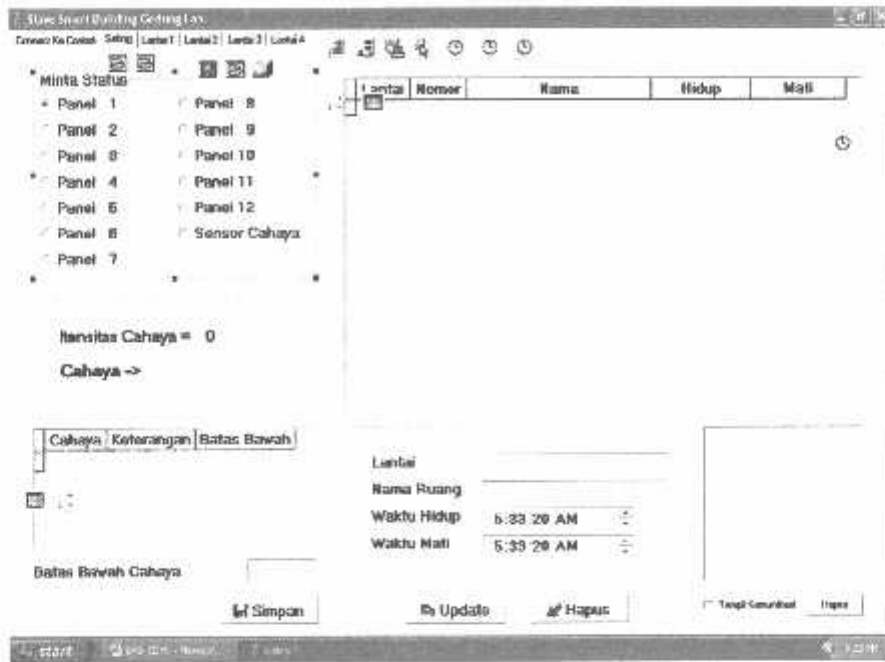
```

Penjelasan :

Masukan data yang diterima *Socket.ReceiveText* kedalam variabel 'x' kemudian data yang ada di dalam var 'x' di cek melalui 'Cek_Terima' setelah itu kosongkan data pada var 'x'.

2. Menyusun Desain Halaman untuk Penyetingan

Menyusun Desain Halaman untuk Penyetingan seperti Gambar berikut



Gambar 3.8. Menyusun Tampilan Form Desain untuk Penyetingan

Untuk mendesain tampilan seperti di atas membutuhkan satu halaman *TabSheet*, dua buah komponen *Panel*, satu buah komponen *RadioGroup*, delapan buah komponen *Label*, satu buah komponen *memo*, satu buah komponen *Button*, Tiga buah komponen *BitBtn*, tiga buah komponen *Edit*, dua buah *DateTimePicker*, satu buah komponen *CheckBox*, dua buah komponen *DBGrid*, dua buah komponen *ADOTable*, dua buah komponen *DataSource*, satu buah *ADOConnection*, Kemudian merubah properti dari masing-masing komponen seperti tabel dibawah.

Tabel 3.3 Merubah isi Properti pada Layar Setting

Komponen	Properti	Isi
<i>PageControl</i> (<i>TabSheet</i>) (Pada palet <i>Win32</i>)	<i>Name</i> <i>Caption</i>	TabSheet1 Seting
Dua Buah <i>Bavel</i> (pada palet <i>Additional</i>)	<i>Name1</i> <i>Name2</i>	<i>Bavel3</i> <i>Bavel5</i>
<i>RadioGroup</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Items</i>	Client Koneksi Panel 1, Panel 2, Panel 3, Panel 4, Panel 5, Panel 6 Panel 7, Panel 8, Panel 9, Panel 10, Panel 11, Panel 12, Sensor Cahaya.
<i>Timer1</i> (Pada palet <i>System</i>)	<i>Name</i> <i>Enable</i> <i>Interval</i>	<i>Timer1</i> <i>True</i> <i>1000ms</i>
<i>Label</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label21 Cahaya ->
<i>Label</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label22 Batas Bawah Cahaya
<i>Label</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label23 Itensitas Cahaya =
<i>Label</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label24 0
<i>Label</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label25 Lantai
<i>Label</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label26 Nama Ruang
<i>Label</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label27 Waktu Hidup

<i>Label</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label28 Waktu Mati
<i>Memo</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Lines</i> <i>ScrollBars</i>	<i>Memo1</i> (kosong) ssBoth
<i>Button</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	<i>Button1</i> Hapus
<i>BitBtn</i> (pada palet <i>Additional</i>)	<i>Name</i> <i>Caption</i> <i>Glyph</i>	<i>BitBtn7</i> Hapus Klik pada tiga titik kemudian load gambar icon dengan format bitmaps yang akan digunakan.
<i>BitBtn</i> (pada palet <i>Additional</i>)	<i>Name</i> <i>Caption</i> <i>Glyph</i>	<i>BitBtn8</i> Update Klik pada tiga titik kemudian load gambar icon dengan format bitmaps yang akan digunakan.
<i>BitBtn</i> (pada palet <i>Additional</i>)	<i>Name</i> <i>Caption</i> <i>Glyph</i>	<i>BitBtn9</i> Simpan Klik pada tiga titik kemudian load gambar icon dengan format bitmaps yang akan digunakan.
<i>Edit</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Text</i>	<i>Edit10</i> (kosong)
<i>Edit</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Text</i>	<i>Edit11</i> (kosong)
<i>Edit</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Text</i>	<i>Edit12</i> (kosong)
<i>DateTimePicker</i> (Pada palci Win32)	<i>Name</i> <i>Kind</i>	<i>DateTimePicker3</i> dtkTime

<i>DateTimePicker</i> (Pada palet Win32)	<i>Name</i> <i>Kind</i>	<i>DateTimePicker4</i> <i>dtkTime</i>
<i>CheckBox</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	<i>CheckBox1</i> Tampil Komunikasi
<i>DBGrid</i> (Pada palet <i>Data Control</i>)	<i>Name</i> <i>DataSource</i> <i>ReadOnly</i>	<i>DBGrid1</i> <i>DataSource1</i> <i>True</i>
<i>DBGrid</i> (Pada palet <i>Data Control</i>)	<i>Name</i> <i>DataSource</i> <i>ReadOnly</i>	<i>DBGrid2</i> <i>DataSource2</i> <i>True</i>
<i>ADOTable</i> (Pada palet <i>ADO</i>)	<i>Name</i> <i>Conection</i> <i>Table Name</i> <i>Aktif</i>	<i>ADOTable1</i> <i>ADOConnection1</i> <i>T Timer</i> <i>False</i>
<i>ADOTable</i> (Pada palet <i>ADO</i>)	<i>Name</i> <i>Conection</i> <i>Table Name</i> <i>Aktif</i>	<i>ADOTable2</i> <i>ADOConnection1</i> <i>T Cahaya</i> <i>False</i>
<i>DataSource</i>	<i>Name</i> <i>Enabled</i> <i>DataSet</i> <i>AutoEdit</i>	<i>DataSource1</i> <i>True</i> <i>ADOTable1</i> <i>True</i>
<i>DataSource</i>	<i>Name</i> <i>Enabled</i> <i>DataSet</i> <i>AutoEdit</i>	<i>DataSource2</i> <i>True</i> <i>ADOTable2</i> <i>True</i>
<i>ADOConnection</i>	<i>Name</i> <i>Provider</i>	<i>ADOConnection1</i> <i>Microsoft.Jet.OLEDB.4.0</i>

- Memasukan Program berikut kedalam *Timer1* pada *Event OnTimer* :
Timer1timer

```

procedure TForm_Utama.Timer1Timer(Sender: TObject);
var
    Status_Panel,
    Status_Cahaya : String;
    nomor_Panel : Byte;
begin
    IF RadioGroup1.ItemIndex = 12
    Then
        Begin
            RadioGroup1.ItemIndex := 0;
            Case Cahaya of
                0: Status_Cahaya:= '8EF10000007F7F7F';
                1: Status_Cahaya:= '8EF10100007E7F7F';
                2: Status_Cahaya:= '8EF10200007D7F7F';
                3: Status_Cahaya:= '8EF10000007F7F7F';
            End;
            ClientSocket1.Socket.SendText(Status_Cahaya);
            IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
            IF CheckBox1.Checked Then memo1.Lines.Add(Status_Cahaya);
            awal := 1;
        End
    Else
        if awal = 0
        then
            RadioGroup1.ItemIndex := RadioGroup1.ItemIndex + 1
        Else
            awal := 0;

    IF RadioGroup1.ItemIndex = 12
    Then
        Begin
            ClientSocket1.Socket.SendText('CEB1');
        End
    Else
        Begin
            nomor_Panel := RadioGroup1.ItemIndex+193;
            Status_Panel := DecToHex(nomor_Panel);
            Status_Panel := Status_Panel + DecToHex(not(nomor_Panel) OR $80);
            ClientSocket1.Socket.SendText(Status_Panel);
            IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
            IF CheckBox1.Checked Then memo1.Lines.Add(Status_Panel);
        End;
    IF panel_Error[RadioGroup1.ItemIndex+1] < 5

```

```

Then
  panel_Error[RadioGroup1.ItemIndex+1] :=
    panel_Error[RadioGroup1.ItemIndex+1] + 1;
end;

```

Penjelasan :

Program yang dimasukan kedalam *Event OnTimer : Timer1timer* ini digunakan untuk menjalankan *RadioGroup1.ItemIndex*, dimana setiap perpindahan dari *RadioGroup1.ItemIndex* program akan mengirimkan data kepada *Computer Control* untuk minta status tiap *Panel*.

- Memasukan Program berikut kedalam *DBGrid2* pada *Event OnCellClick : DBGrid2CellClick*

```

procedure TForm_Utama.DBGrid2CellClick(Column: TColumn);
begin
  Label22.Caption := 'Batas Bawah Cahaya ' + ADOTable2Keterangan.Value;
  Edit10.Text := IntToStr(ADOTable2Batas_Bawah.Value);
  Edit10.SetFocus;
end;

```

Penjelasan:

Program ini akan dijalankan bila *cell* dari *DBGrid2* diklik maka pada *lebel22.caption* akan diisi dengan kata 'Batas Bawah Cahaya' di tambah dengan *ADOTable2Keterangan.Value* dan *Edit10.Text* akan diisi dengan data string pada *ADOTable2Batas_Bawah.Value*.

- Memasukan Program berikut kedalam *DBGrid1* pada *Event OnCellClick : DBGrid1CellClick*

```

procedure TForm_Utama.DBGrid1CellClick(Column: TColumn);
begin
  Edit11.Text := IntToStr(ADOTable1Lantai.Value);
  Edit12.Text := ADOTable1Nama.Value;
  DateTimePicker3.Time := ADOTable1Hidup.Value;
  DateTimePicker4.Time := ADOTable1Mati.Value;
  DateTimePicker3.SetFocus;
end;

```

Penjelasan :

Bila *DBGrid1CellClick* *Edit11.Text* akan diisi data string dari *ADOTable1Lantai.Value*, *Edit12.Text* akan diisi dengan *ADOTable1Nama.Value* kemudian *DateTimePicker3.Time* akan diisi dengan *ADOTable1Hidup.Value* dan *DateTimePicker4.Time* akan diisi dengan *ADOTable1Mati.Value*.

Memasukan Program berikut kedalam *BitBtn8* pada **Event OnClick** :
BitBtn8Click.

```
procedure TForm_Utama.BitBtn8Click(Sender: TObject);
begin
  IF ADOTable1.Locate('nama',Edit12.Text,[])
  Then
    Begin
      ADOTable1.Edit;
      ADOTable1Hidup.Value := DateTimePicker3.Time;
      ADOTable1Mati.Value := DateTimePicker4.Time;
      ADOTable1.Post;
    End;
end;
```

Penjelasan:

Program di atas berfungsi meng-edit yang ada pada *ADOTableHidup.Value* dan *ADOTable1Mati.Value*.

- Memasukan Program berikut kedalam *BitBtn9* pada **Event OnClick** :
BitBtn9Click.

```
procedure TForm_Utama.BitBtn9Click(Sender: TObject);
begin
  IF StrToIntDef(Edit10.Text,0) <= 4000
  Then
    Begin
      ADOTable2.Edit;
      ADOTable2Batas_Bawah.Value := StrToIntDef(Edit10.Text,0);
      ADOTable2.Post;
      Edit10.Clear;
      ADOTable2.First;
      B_Terang := ADOTable2Batas_Bawah.Value;
```

```

    ADOTable2.Next;
    B_Sedang := ADOTable2Batas_Bawah.Value;
    Label22.Caption := 'Batas Bawah Cahaya ' + ADOTable2Keterangan.Value;
    Edit10.Text := IntToStr(ADOTable2Batas_Bawah.Value);
  End
Else
  MessageDlg('besarnya harus antara 0 sampai 4000',mtError,[mbOk],0);
end;

```

Penjelasan :

OnClick BitBtn9Click artinya bila *BitBtn9* diklik maka program yang ada pada *BitBtn9Click* akan dijalankan. Yaitu jika data yang dimasukan pada *edit10.text* <= 4000 maka dapat melakukan perubahan data pada *ADOTable2*, jika data yang dimasukan lebih dari 4000 maka akan muncul pesan besarnya harus antara 0 sampai 4000. dan data akan diisikan dengan *default nol*.

- Memasukan Program berikut kedalam *BitBtn7* pada *Event OnClick : BitBtn7Click*.

```

procedure TForm_Utama.BitBtn7Click(Sender: TObject);
begin
  IF ADOTable1.RecordCount > 0
  Then
    Begin
      ADOTable1.Delete;
      Edit11.Clear;
      Edit12.Clear;
    End;
end;

```

Penjelasan :

Program di atas mempunyai fungsi untuk menghapus data yang ada pada *ADOTable1*.

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lantai1	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3118</i> Lab. Sistem Tenaga 3118	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3118	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3103</i> Kalab. Sistem Tenaga 3103	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3103	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3102</i> Ruang Kontrol 3102	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3102	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3101</i> Gudang 3101	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3101	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3119</i> Perpustakaan 3119	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3119	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2106</i> WC 1 2106	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2106	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2107</i> WC 2 2107	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2107	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2108</i> WC 3 2108	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 2108	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2105</i> WC 4 2105	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2105a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 2105b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2109</i> WC Luar 2109	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2109	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3104</i> Tangga 3104	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3104a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3104b	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2110</i> Elite 2110	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2110	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1104</i> Ruang Seminar 1104	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1104	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1103</i> Ruang Dosen 1103	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1103	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1102</i> Ruang Korbid 1102	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1102	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1107</i> SSDP1 1107	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1117</i> HALL1 1117	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107d	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107e	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107f	(kosong)	(kosong)

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1107h	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1118</i> HALL2 1118	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118d	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118e	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118f	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118g	(kosong)	(kosong)

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1118h	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1119</i> Lab. Konversi Energi Listrik 1119	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1119	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1101</i> Kalab. Konversi Energi Listrik 1101	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1101	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2118</i> Lab. Transmisi dan Distribusi 2118	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2118	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2101</i> Kalab. Transmisi dan Distribusi 2101	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 2101	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 2109</i> Worksop 2109	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2109	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2102</i> Kalab. Woksop 2102	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 2102	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1106</i> Teras Depan 1106	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 1106a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1106b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1106c	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3105</i> Teras Belakang 3105	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3105	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3106</i> Teras Kiri 3106	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3106a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3106b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2104</i> Teras Kanan 2104	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2104	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1105</i> Depan Kiri 1105	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1105a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1105b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2103</i> Depan Kanan 2103	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2103a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2103b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2103c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Image1	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Image9	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_11	(kosong)	(kosong)

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_21	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_31	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_Kosong	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	TL_Off_Off	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	TL_On_Off	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	TL_On_On	(kosong)	(kosong)
<i>Label</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label9 Cahaya -> Sedang	(kosong)	(kosong)
<i>Button</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Button2 Gelap	OnClick	Button2Click
<i>Button</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Button3 Sedang	OnClick	Button3Click
<i>Button</i> (pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Button4 Terang	OnClick	Button4Click

- Memasukan Program berikut kedalam *Tombol_1117* pada **Event OnClick** :
Tombol_1117Click.

```

procedure TForm_Utama.Tombol_1117Click(Sender: TObject);
Var
    No_Input : Integer;
    No_Panel : Byte;
    x,y,a,datanya : string;
begin
    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
    Dua_Relay(No_Panel,No_Input);
    X := inttostr(no_panel);
    y := inttostr(no_input);
    a := (x+'-'+y);
    ClientSocket1.Socket.SendText(a);
end;

```

Penjelasan:

Program ini digunakan untuk mengisi data *No_Panel* dan *No_input* yang berasal dari *Tag Panel* yang di klik. Kemudian data tersebut dimasukan kedalam subprogram *Dua_relay* yang berfungsi untuk mengaktifkan atau mematikan lampu. Selain itu pada saat *Panel* di klik *ClientSocket1* akan mengirimkan data yang berada di dalam variabel 'a' ke *Computer Control* yang berfungsi sebagai perintah untuk melakukan perubahan status *Panel*.

- Memasukan Program berikut kedalam *Tombol_1101* pada **Event OnClick** :
Tombol_1101Click.

```

procedure TForm_Utama.Tombol_1101Click(Sender: TObject);
Var
    No_Input : Integer;
    No_Panel : Byte;
    x,y,a: string;
begin
    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
    Satu_Relay(No_Panel,No_Input);
    x := inttostr(no_panel);
    y := inttostr(no_input);

```

```

a := (x+','+y);
ClientSocket1.Socket.SendText(a);
end;

```

Penjelasan:

Program ini digunakan untuk mengisi data No_Panel dan No_input yang berasal dari *Tag Panel* yang di klik. Kemudian data tersebut dimasukan kedalam subprogram Satu_relay yang berfungsi untuk mengaktifkan lampu atau memadamkan. Selain itu pada saat Panel di klik *ClientSocket1* akan mengirimkan data yang berada di dalam variabel 'a' ke *Computer Control* yang berfungsi sebagai perintah untuk melakukan perubahan status *Panel*.

- Memasukan Program berikut kedalam *Tombol_1107* pada *Event OnClick* :
Tombol_1107Click

```

procedure TForm_Utama.Tombol_1107Click(Sender: TObject);
Var
    No_Input : Integer;
    No_Panel : Byte;
    Status_Cahaya : Integer;
    a,x,y : string;
begin
    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
    Status_Cahaya := Cahaya;
    Cahaya := 0;
    Satu_Relay(No_Panel,No_Input);
    Cahaya := Status_Cahaya;
    x := inttostr(no_panel);
    y := inttostr(no_input);
    a := (x+','+y);
    ClientSocket1.Socket.SendText(a);
end;

```

Penjelasan:

Program ini digunakan untuk mengisi data No_Panel dan No_input yang berasal dari *Tag Panel* yang di klik. Kemudian data tersebut dimasukan kedalam

subprogram *Satu_relay* yang berfungsi untuk mengaktifkan atau mematikan SSDP. Selain itu pada saat Panel di klik *ClientSocket1* akan mengirimkan data yang berada di dalam variabel 'a' ke *Computer Control* yang berfungsi sebagai perintah untuk melakukan perubahan status *Panel*.

- Memasukan Program berikut kedalam *Tombol_3118* pada **Event OnContextPopup** : Tombol 3118ContextPopup

```
procedure TForm_Utama.Tombol_3118ContextPopup(Sender: TObject;
    MousePos: TPoint; var Handled: Boolean);
Var
    No_Input : Integer;
    No_Panel : Byte;
begin
    Panel3.BringToFront;
    Panel3.Visible := True;

    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := (Sender As TPanel).Tag;

    IF ADOTable1.Locate('nomor',No_Input,[])
    Then
        Begin
            Edit7.Text := IntToStr(ADOTable1.Lantai.Value);
            Edit8.Text := ADOTable1.Nama.Value;
            Edit9.Text := IntToStr(ADOTable1.Nomor.Value);
            DateTimePicker1.Time := ADOTable1.Hidup.Value;
            DateTimePicker2.Time := ADOTable1.Mati.Value;
        End
    Else
        Begin
            DateTimePicker1.Time := Time;
            DateTimePicker2.Time := Time;
        End;
    IF No_Panel <=3 Then Edit7.Text := '1'
    Else IF No_Panel <=6 Then Edit7.Text := '2'
    Else IF No_Panel <=9 Then Edit7.Text := '3'
    Else Edit7.Text := '4';
    Edit8.Text := (Sender As TPanel).Caption;
    Edit9.Text := IntToStr(No_Input);
end;
```

Penjelasan:

Program ini digunakan untuk mengisi data ke tabel dimana isi dari tabel berfungsi untuk melakukan penjadwalan pengaktifan lampu maupun pemadaman lampu. Tampilan program ini akan dimunculkan bila menekan tombol mouse kanan yang diletakan pada *Panel*.

- Memasukan Program berikut kedalam Tombol_3118 pada *Event* Tombol_3118MouseDown

```
procedure TForm_Utama.Tombol_3118MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  (Sender As TPanel).BevelInner := bvLowered;
  (Sender As TPanel).BevelOuter := bvLowered;
end;
```

Penjelasan:

Program ini berfungsi untuk membentuk garis siku di dalam dan luar pada panel yang digunakan

- Memasukan Program berikut kedalam Tombol_3118 pada *Event* Tombol_3118MouseUp

```
procedure TForm_Utama.Tombol_3118MouseUp(Sender: TObject; Button:
  TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  (Sender As TPanel).BevelInner := bvRaised;
  (Sender As TPanel).BevelOuter := bvRaised;
end;
```

Penjelasan:

Program ini berfungsi untuk membentuk garis siku dalam dan luar pada panel yang digunakan

- Memasukan Program berikut kedalam *Button2* pada *Event* *OnClick* : *Button2Click*

```

procedure TForm_Utama.Button2Click(Sender: TObject);
begin
    Cahaya := 0;
    ClientSocket1.Socket.SendText('e');
end;

```

Penjelasan :

Jika *Button2* diklik maka program akan mengisi data nol pada Cahaya dan akan mengirimkan huruf 'e' yang bertipe *string* melalui *Socket*.

- Memasukan Program berikut kedalam *Button3* pada *Event OnClick* :

```

procedure TForm_Utama.Button3Click(Sender: TObject);
begin
    Cahaya := 1;
    ClientSocket1.Socket.SendText('k');
end;

```

Penjelasan :

Jika *Butto3* diklik maka program akan mengisi data satu pada Cahaya dan akan mengirimkan huruf 'k' yang bertipe *string* melalui *Socket*.

- Memasukan Program berikut kedalam *Button4* pada *Event OnClick* :

```

procedure TForm_Utama.Button4Click(Sender: TObject);
begin
    Cahaya := 2;
    ClientSocket1.Socket.SendText('o');
end;

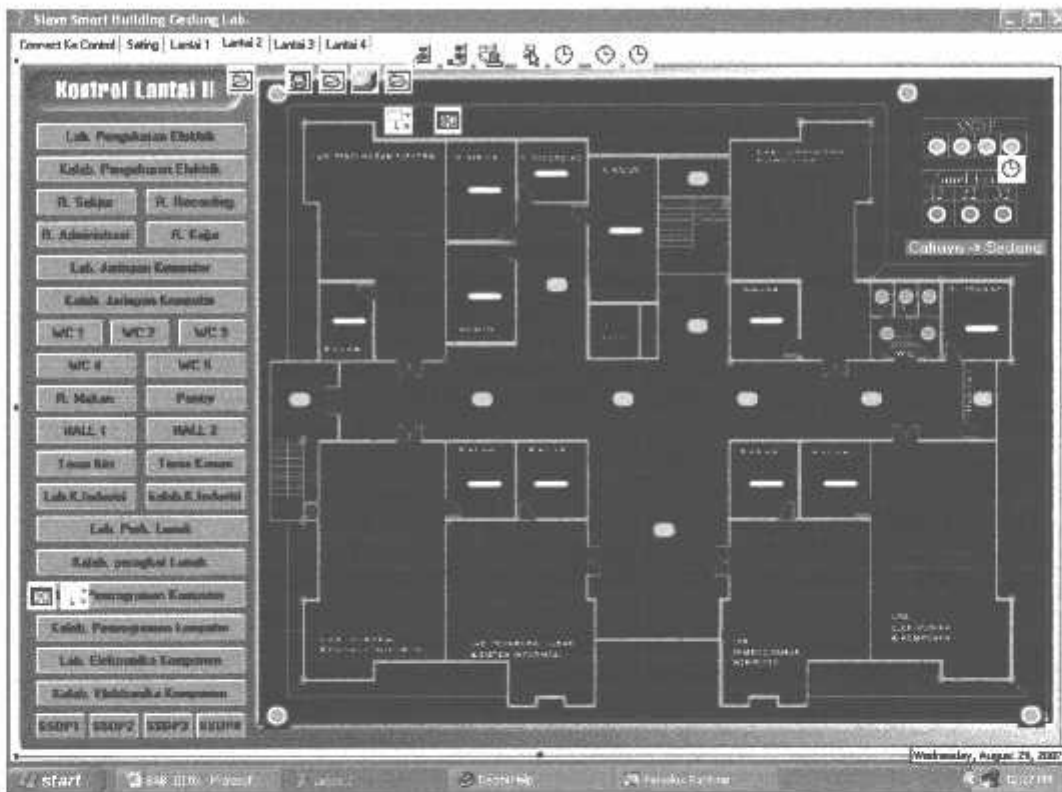
```

Penjelasan :

Jika *Button2* diklik maka program akan mengisi data dua pada Cahaya dan akan mengirimkan huruf 'o' yang bertipe *string* melalui *Socket*.

3. Menyusun Desain Tampilan Halaman lantai Dua

Menyusun Desain Tampilan Halaman lantai Dua seperti pada gambar dibawah



Gambar 3.10. Desain Tampilan Halaman lantai Dua

Untuk mendesain tampilan seperti gambar di atas membutuhkan satu halaman *TabSheet*, tigapuluh satu buah komponen *Panel*, satu buah komponen *Label*, empatpuluh komponen *image*, Kemudian merubah properti dari masing-masing komponen seperti tabel berikut

Tabel 3.5 Merubah isi Properti dan Memasukan Event
pada Halaman Lantai 2

Komponen	Properti	Isi	Event	isi
PageControl (TabSheet) (Pada palet Win32)	Name Caption	TabSheet3 Lantai2	(kosong)	(kosong)
Image (Pada palet Additional)	Name	Lantai2	(kosong)	(kosong)
Panel (Pada palet standard)	Name Caption Tag	Tombol_3219 Lab. Pengukuran Elektrik 3219	OnClick OnContextPopup OnMouseDown OnMouseUp	Tombol_1117Click Tombol_3118ContextPopup Tombol_3118MouseDown Tombol_3118MouseUp
Image (Pada palet Additional)	Name	Lampu_3219	(kosong)	(kosong)
Panel (Pada palet standard)	Name Caption Tag	Tombol_3205 Kalab. Pengukuran Elektrik 3205	OnClick OnContextPopup OnMouseDown OnMouseUp	Tombol_1101Click Tombol_3118ContextPopup Tombol_3118MouseDown Tombol_3118MouseUp
Image (Pada palet Additional)	Name	Lampu_3205	(kosong)	(kosong)
Panel (Pada palet standard)	Name Caption Tag	Tombol_3203 R. Sekjur 3203	OnClick OnContextPopup OnMouseDown OnMouseUp	Tombol_1101Click Tombol_3118ContextPopup Tombol_3118MouseDown Tombol_3118MouseUp
Image	Name	Lampu_3203	(kosong)	(kosong)

(Pada palet <i>Additional</i>)				
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3202</i> R. Recording 3202	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3202	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3204</i> R. Administrasi 3204	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3204	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3201</i> R. Kujur 3201	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu 3201	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2219</i> Lab. Jaringan Komputer 2119	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2119	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2204</i> Kalab. Jaringan Komputer 2204	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2204	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2206</i> WC 1 2206	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2206	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2207</i> WC 2 2207	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2207	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2208</i> WC 3 2208	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2208	(kosong)	(kosong)
<i>Panel</i> (Pada palet)	<i>Name</i> <i>Caption</i>	<i>Tombol_2209</i> WC 4	<i>OnClick</i> <i>OnContextPopup</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i>

<i>standard)</i>	<i>Tag</i>	2209	<i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2209	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard)</i>	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2210</i> WC 5 2210	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2210	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard)</i>	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2205</i> R. Makan 2205	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2205	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard)</i>	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2212</i> R. Pantry 2212	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2212	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard)</i>	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1204</i> HALL1 1204	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1204a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1204b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1204c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1204d	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1204e	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2203</i> HALL2 2203	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2203a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2203b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2203c	(kosong)	(kosong)

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2203d	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1203</i> Teras Kiri 1203	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1203a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1203b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2211</i> Teras Kanan 2211	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2211a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2211b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1219</i> Lab.K.Industri 1219	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1219	(kosong)	(kosong)

<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 1201</i> Kalab.K. Industri 1201	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_1201	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 1218</i> Lab. Perangkat Lunak 1218	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_1218	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1202</i> Kalab. Perkangkat Lunak 1202	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_1202	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 2217</i> Lab. Pemrograman Komputer 2217	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_2217	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2201</i> Kalab. Pemrograman Komputer 2201	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2201	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2218</i> Lab. Elektronika Komponen 2218	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2218	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2202</i> Kalab. Elektronika Komponen 2202	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2202	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1205</i> SSDP1 1205	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1205	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1206</i> SSDP2 1206	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1206	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3206</i> SSDP3 3206	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3206	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3207</i> SSDP4 3207	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3207	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Image6	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Image2	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_12	(kosong)	(kosong)

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_22	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Panel_32	(kosong)	(kosong)
<i>Label</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i>	Label10 Cahaya -> Sedang	(kosong)	(kosong)

Untuk program pada setiap panel karena memiliki program yang sama sehingga program dijadikan satu pada *Events*, jika pada *Events OnClick* – *Tombol_1117Click* maka pada setiap panel yang *Events OnClick*-nya sama akan menjalankan program yang ada didalam *Events OnClick* tersebut.

5. Menyusun Desain Tampilan Halaman Lantai Tiga

Menyusun Desain Tampilan Halaman lantai Tiga seperti pada Gambar 3.9.
berikut:

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lantai3	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1317</i> Lab. Elektronika Analog 1317	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1317	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1301</i> Kalab. Elektronika Analog 1301	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1301	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1318</i> Lab. Dasar Pemrograman Komputer 1318	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1318	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1305</i> Kalab. Dasar Pemrograman Komputer 1305	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lantai3	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 1317</i> Lab. Elektronika Analog 1317	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol 1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1317	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1301</i> Kalab. Elektronika Analog 1301	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1301	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 1318</i> Lab. Dasar Pemrograman Komputer 1318	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol 1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1318	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1305</i> Kalab. Dasar Pemrograman Komputer 1305	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1305	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1303</i> Tangga 1303	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1303a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1303b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3317</i> R. SIM 3317	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3317	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3316</i> R. DIII 3316	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3316	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2308</i> Wudu 1 2308	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2308	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2309</i> Wudu 2 2309	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2309	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3311</i> Mushola 3311	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3311	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3305</i> WC 1 3305	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3305	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3306</i> WC 2 3306	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3306	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3307</i> WC 3 3307	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3307	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3308</i> WC 4 3308	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3308a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3308b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1304</i> SSDP1 1304	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1304	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2305</i> SSDP2 2305	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2305	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2306</i> SSDP3 2306	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2306	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2307</i> SSDP4 2307	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2307	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3312</i> SSDP5 3312	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1107Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3312	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2304</i> HALL1 2304	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2304a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2304b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2304c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2304d	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3309</i> HALL2 3309	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3309a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3309b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3309c	(kosong)	(kosong)
<i>Panel</i>	<i>Name</i>	<i>Tombol_2319</i>	<i>OnClick</i>	<i>Tombol_1117Click</i>

(Pada palet standard)	<i>Caption</i> <i>Tag</i>	Lab. Elektronika Digital 2319	<i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_2319	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2303</i> Kalab. Eletronika Digital 2303	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_2303	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2318</i> Lab. Pemeliharaan Perbaikan Elka 2318	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_2318	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2302</i> Kalab. Pemeliharaan Perbaikan Elka 2302	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_2302	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 3319</i> Lab. Multimedia 3319	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol 1117Click</i> <i>Tombol 3118ContextPopup</i> <i>Tombol 3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3319	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 3302</i> Kalab. Multimedia 3302	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol 3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3302	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol 3301</i> R. Korbid 3301	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol 3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3301	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3318</i> R. Sidang I 3318	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol 3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3318	(kosong)	(kosong)

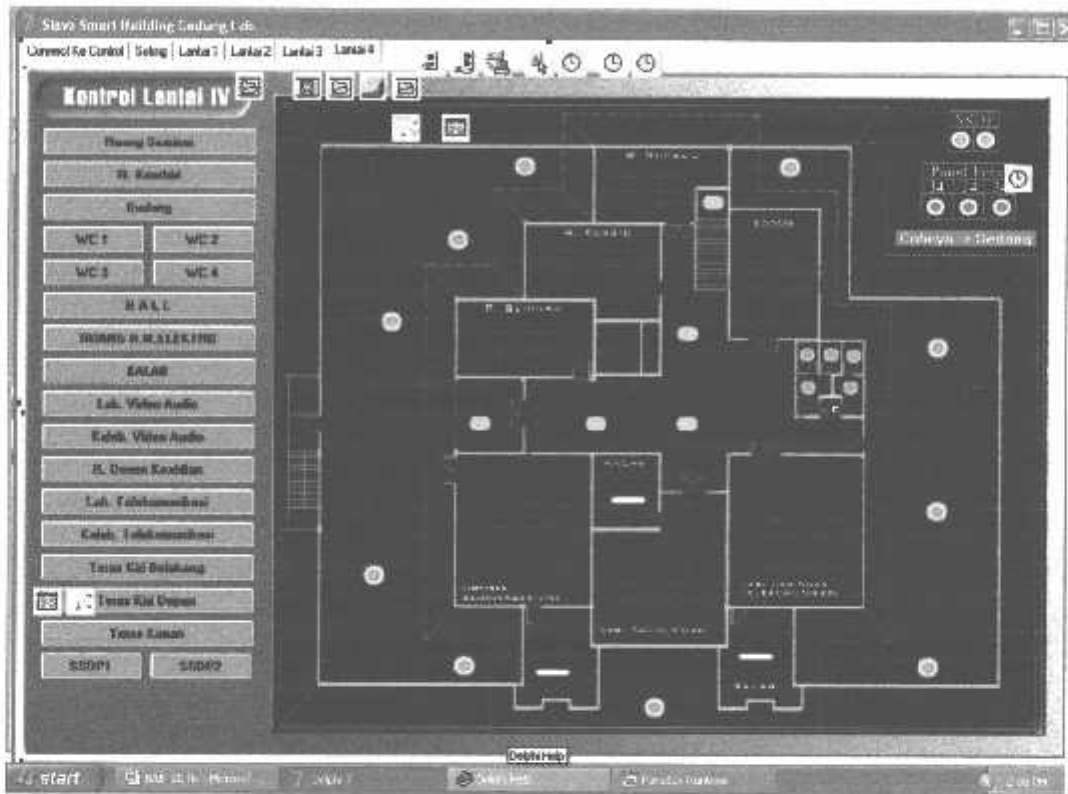
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3310</i> R. Sidang II 3310	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3310a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3310b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3310c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3310d	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1302</i> Bel. Kiri 1302	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1302	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3304</i> Bel. Kanan 3304	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3304	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2301</i> Depan Kiri 2301	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2301	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3303</i> Depan Kanan 3303	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3303	(kosong)	(kosong)

Karena Program pada beberapa panel memiliki program yang sama sehingga program dijadikan satu pada *Events*, jika suatu *Panel* tertentu meskipun bukan *Tombol_1117* pada *Events*-nya *OnClick* – *Tombol_1117Click* maka panel yang *Events OnClick*-nya *Tombol_1117Click* akan menjalankan program yang ada didalam *Events OnClick* tersebut.

6. Menyusun Desain Tampilan Halaman Lantai Empat

Menyusun Desain Tampilan Halaman lantai Empat seperti pada gambar dibawah



Gambar 3.12. Desain Tampilan Halaman Lantai Empat

Untuk mendesain tampilan seperti gambar di atas membutuhkan satu halaman *TabSheet*, duapuluh buah komponen *Panel*, satu buah komponen *Label*, tigapuluh delapan komponen *image*. Kemudian merubah properti dari masing-masing komponen seperti tabel dibawah.

Tabel 3.7 Merubah isi Properti dan Memasukan Event
pada Halaman Lantai 4

Komponen	Properti	Isi	Event	isi
<i>PageControl</i> (<i>TabSheet</i>) (Pada palet <i>Win32</i>)	<i>Name</i> <i>Caption</i>	TabSheet5 Lantai4	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lantai4	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1417</i> R. Seminar 1417	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1417	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1418</i> R. Korbid 1418	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1418	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1419</i> Gudang 1419	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1419	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2401</i> WC 1 2401	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2401	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2402</i> WC 2 2402	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2402	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2403</i> WC 3 2403	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2403	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2404</i> WC 4 2404	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2404a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2404b	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1402</i> HALL 1402	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1402a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1402b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1402c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1402d	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1402e	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3419</i> RUANG HM ELEKTRO 3419	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3419	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3403</i> KALAB 3403	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3403	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3418</i> Lab. Vidio Audio 3418	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3418	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3402</i> Kalab. Vidio Audio 3402	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3402	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2418</i> R. Dosen Keahlian 2418	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>

<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2418	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2419</i> Lab. Telekomunikasi 2419	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2419	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2406</i> Kalab. Telekomunikasi 2406	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2406	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_1401</i> Teras Kiri Belakang 1401	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1401a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1401b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_1401c	(kosong)	(kosong)

<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3401</i> Teras Kiri Depan 3401	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1117Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3401a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3401b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_3401c	(kosong)	(kosong)
<i>Panel</i> (Pada palet <i>standard</i>)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_2405</i> Teras Kanan 2405	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2405a	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2405b	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2405c	(kosong)	(kosong)
<i>Image</i> (Pada palet <i>Additional</i>)	<i>Name</i>	Lampu_2405d	(kosong)	(kosong)

<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3404</i> SSDP1 3404	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_3404	(kosong)	(kosong)
<i>Panel</i> (Pada palet standard)	<i>Name</i> <i>Caption</i> <i>Tag</i>	<i>Tombol_3405</i> SSDP2 3405	<i>OnClick</i> <i>OnContextPopup</i> <i>OnMouseDown</i> <i>OnMouseUp</i>	<i>Tombol_1101Click</i> <i>Tombol_3118ContextPopup</i> <i>Tombol_3118MouseDown</i> <i>Tombol_3118MouseUp</i>
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Lampu_3404	(kosong)	(kosong)
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Image4	(kosong)	(kosong)
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Image8	(kosong)	(kosong)
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Panel_14	(kosong)	(kosong)
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Panel_24	(kosong)	(kosong)
<i>Image</i> (Pada palet Additional)	<i>Name</i>	Panel_34	(kosong)	(kosong)

<i>Label</i>	<i>Name</i>	<i>Label12</i>		
(Pada palet <i>Standard</i>)	<i>Caption</i> <i>Color</i>	Cahaya -> Sedang Red	(kosong)	(kosong)

Untuk memperpendek program dari *Panel* yang memiliki Program yang sama, program dijadikan satu pada *Events*, jika suatu *Panel* tertentu misalkan bukan *Tombol_1117* tetapi pada *Events OnClick-nya – Tombol_1117Click* maka panel yang *Events OnClick-nya Tombol_1117Click* akan menjalankan program yang ada didalam *Events OnClick* dari *Tombol_1117*.

Rancangan program untuk membaca status tiap panel dari jawaban yang diterima dari *Computer Control* melalui *socket* yang kemudian akan ditampilkan pada monitor adalah sebagai berikut:

```

Procedure TForm_Utama.Cek_Terima(datanya : String);
Var
  D1,D2,D3,
  D1_Inv,D2_Inv,D3_Inv : Byte;
  Sinar : real;
Begin
  Timer1.Enabled := False;
  IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
  IF CheckBox1.Checked Then Memo1.Lines.Add(datanya);
  D1 := Hexbin(Copy(datanya,1,2));
  D2 := Hexbin(Copy(datanya,3,2));
  D3 := Hexbin(Copy(datanya,5,2));
  D1_Inv := Hexbin(Copy(datanya,7,2));
  D2_Inv := Hexbin(Copy(datanya,9,2));
  D3_Inv := Hexbin(Copy(datanya,11,2));
  IF ( ((not(D1) AND $7F) = D1_Inv) AND
    ((not(D2) AND $7F) = D2_Inv) AND
    ((not(D3) AND $7F) = D3_Inv) )
  Then
    Begin
      panel_Error[RadioGroup1.ItemIndex+1] := 0;
      if RadioGroup1.ItemIndex = 12
      Then
        Begin
          IF copy(datanya,1,2) = '3B'
          Then Sinar := 0
          Else IF copy(datanya,1,2) = '2A'

```

Penjelasan:

“Datanya” merupakan jawaban *Computer Control* atas pertanyaan dari *Slave Computer* mengenai keadaan status tiap *Panel*. Datanya akan diproses kemudian digunakan untuk menampilkan gambar sebagai indikasi status lampu yang sedang menyala atau padam.

BAB IV

PENGUJIAN RANGKAIAN DAN ANALISA DATA

Untuk mendapatkan hasil yang maksimal setelah melaksanakan perancangan dan pembuatan perangkat, maka perlu dilakukan suatu pengujian terhadap perangkat yang telah dibuat. Pengujian ini bertujuan untuk mengetahui apakah perangkat yang telah dibuat dapat bekerja sesuai yang dengan perencanaan.

Bagian-bagian yang diuji dari perangkat ini adalah :

1. Pengiriman Data dari *Slave Computer* ke *Computer Control* dan sebaliknya.
2. Pengujian Jawaban dari *Computer Control* atas Pertanyaan *Slave Computer* dengan Membandingkan Jawaban *Panel* atas Pertanyaan *Computer Control*.
3. Perubahan Status Lampu pada *Computer Control* dan *Panel* atas Permintaan *Slave Computer* dan mengetahui respon terhadap perubahan status.

4.1. Pengujian Pengiriman Data dari *Slave Computer* ke *Computer Control*

4.1.1. Tujuan

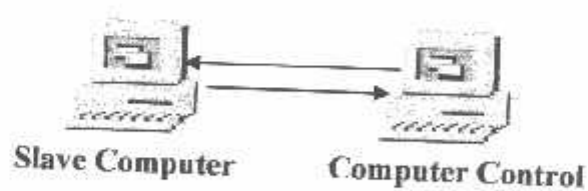
Untuk mengetahui apakah data yang dikirimkan oleh *Slave Computer* ke *Computer Control* atau sebaliknya sesuai dengan data yang aslinya.

4.1.2. Peralatan Yang Digunakan

1. *Slave Computer.*
2. *Computer Control.*
3. *Ethernet Card.*
4. Kabel UTP (*Unshielded Twister Pair*).

4.1.3. Langkah Pengujian

1. Membuka program Delphi pada dua buah Komputer dan menjalankan program aplikasi.
2. Menghubungkan kabel UTP pada *Slave Computer* ke *Computer Control*.
3. Seting alamat IP dengan IP lokal 10.0.0.1 untuk *Computer Control* dan 10.0.0.2 untuk *Slave Computer*.
4. *Connect* kan *server socket* (antara *Computer Control* dan *Slave Computer*).
5. Mengirimkan data dari *Slave Computer* dan melihat data yang diterima *Computer Control*.
6. Mengirimkan data dari *Computer Control* dan melihat data yang diterima *Slave Computer*.



Gambar 4.1 Diagram Blok Pengujian Pengiriman Data.

4.1.4. Hasil Pengujian

Hasil pengujian Pengiriman Data dengan mengirim kata “Data1” dari *Slave Computer* dan Mengirim kata “Data2” dari *Computer Control* adalah sebagai berikut:

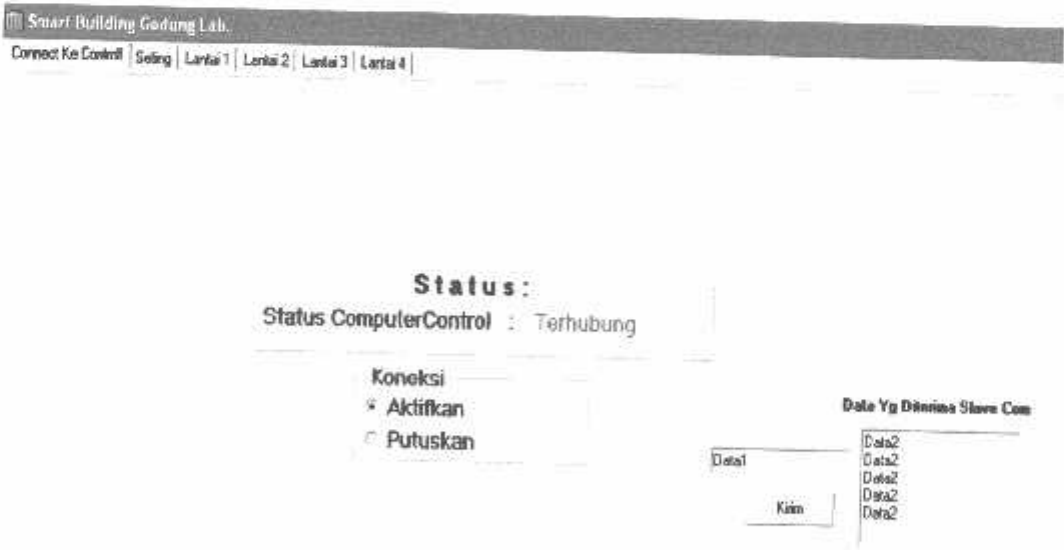
Sebelum melakukan pengiriman data Perlu menambahkan satu buah komponen *Edit*, satu buah komponen *Button*, satu buah komponen *Memo* dan satu buah komponen *Label* pada kedua Buah Komputer. Kemudian *double* klik pada *Button*

Dan mengetikkan program berikut

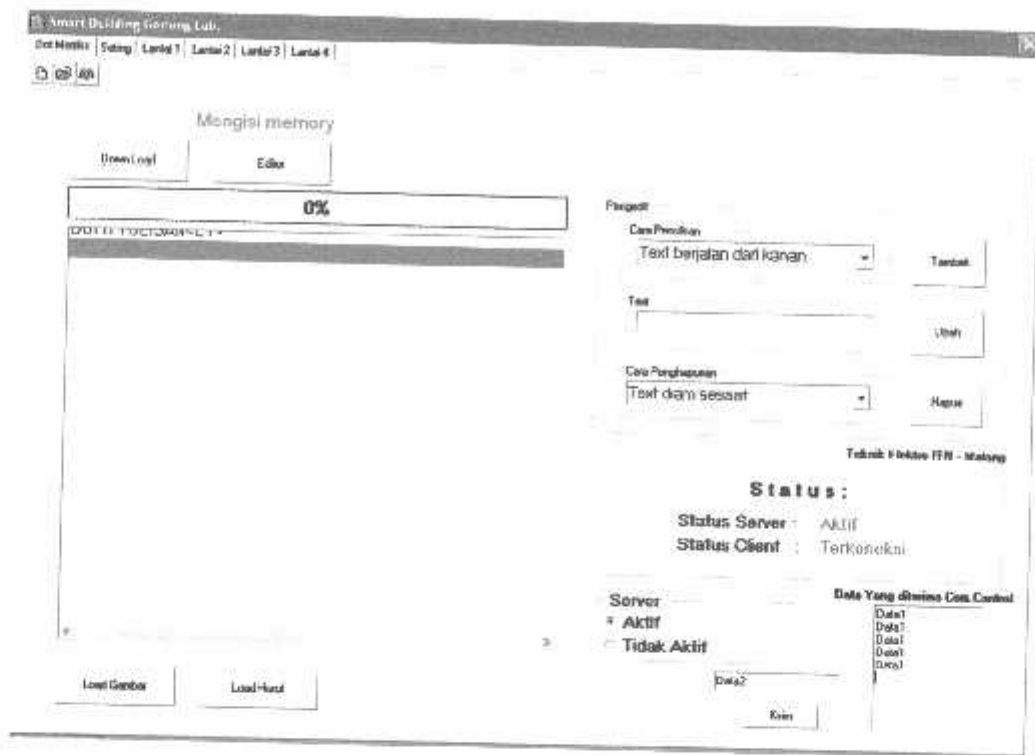
```
ClientSocket1.Socket.SendText(Edit14.Text);
```

Dengan mengetikkan Program di atas data yang ada pada *edit14.text* akan dikirimkan melalui *socket*. Untuk menampilkan data yang diterima perlu mengetikkan program berikut di dalam komponen *ServerSocket1* pada *event OnClientRead* *ServerSocket1ClientRead*:

```
Memo3.Lines.Add(Socket.ReceiveText);
```



Gambar 4.2 Data yang Dikirim dan Diterima Oleh *Slave Computer*



Gambar 4.3 Data yang Dikirim dan Diterima Oleh *Computer Control*

4.2. Pengujian Jawaban dari *Computer Control* atas Pertanyaan *Slave Computer* dengan Membandingkan Jawaban *Panel* atas Pertanyaan *Computer Control*.

4.2.1. Tujuan

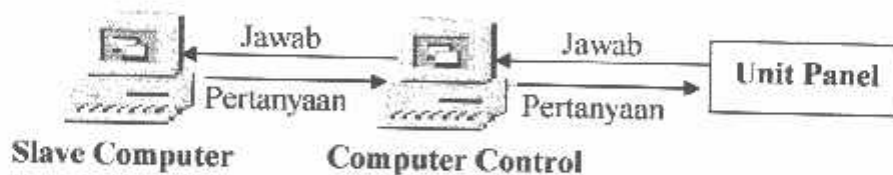
Untuk mengetahui apakah data yang dikirimkan oleh *Computer Control* yang berasal dari *Panel* sesuai dengan pertanyaan dari *Slave Computer*.

4.2.2. Peralatan yang Digunakan

1. *Unit Panel*
2. *Slave Computer*.
3. *Computer Control*.
4. *Ethernet Card*.
5. Kabel UTP (*Unshielded Twister Pair*).

4.2.3. Langkah Pengujian

1. Menghubungkan Kabel Serial antara *Computer Control* dan *Panel*.
2. Menghubungkan kabel UTP pada *Slave Computer* ke *Computer Control*.
3. Membuka program Delphi pada dua buah Komputer dan menjalankan program aplikasi.
4. Pastikan *Computer Control* telah terhubung pada *Unit Panel* dan dapat berkomunikasi.
5. Seting alamat IP dengan IP lokal 10.0.0.1 untuk *Computer Control* dan 10.0.0.2 untuk *Slave Computer*.
6. *Connect* kan *server socket* (antara *Computer Control* dan *Slave Computer*). Bila sudah *Connect*, maka
7. *Slave Computer* akan Mengirimkan data yang akan digunakan sebagai pertanyaan keadaan status tiap *Panel* pada *Computer Control*.
8. Bandingkan Jawaban yang diterima *Slave Computer* dari *Computer Control* dengan jawaban *Panel* yang diterima oleh *Computer Control*.



Gambar 4.4 Diagram Blok Pengujian Jawaban

4.2.4. Hasil Pengujian

Tabel 4.1 Perbandingan Hasil Pengujian Data yang Diterima *Computer Control* dari *Panel* dan Data yang diterima *Slave Computer* dari *Computer Control* pada saat Status *Panel* tidak ada Perubahan.

Pertanyaan	Jawaban Yang Diterima	
	<i>Computer Control</i>	<i>Slave Computer</i>
C1BE	1400066B7F79	1400066B7F79
C2BD	582006275F79	582006275F79
C3BC	0400007B7F7F	0400007B7F7F
C4BB	4400003B7F7F	1400066B7F79
C5BA	633C001C437F	633C001C437F
C6B9	1200006D7F7F	1200006D7F7F
C7B8	0000007F7F7F	0000007F7F7F
C8B7	2000065F7F79	2000065F7F79
C9B6	0040007F3F7F	7F7F0F000070
CAB5	0000007F7F7F	0000007F7F7F
CBB4	4100003E7F7F	4100003E7F7F
CCB3	0A6000751F7F	0A6000751F7F
8EF10000007F7F7F	-	-

Dengan Pertanyaan yang sama Jawaban yang diterima pun juga harus benar-benar sama. Bila tidak maka tampilan lampu tidak akan sesuai dengan keadaan sesungguhnya.

4.3. Perubahan Status Panel atas Permintaan *Slave Computer* dan mengetahui respon terhadap perubahan status..

4.3.1. Tujuan

Untuk membuktikan apakah *Slave Computer* dapat melakukan Perintah perubahan.

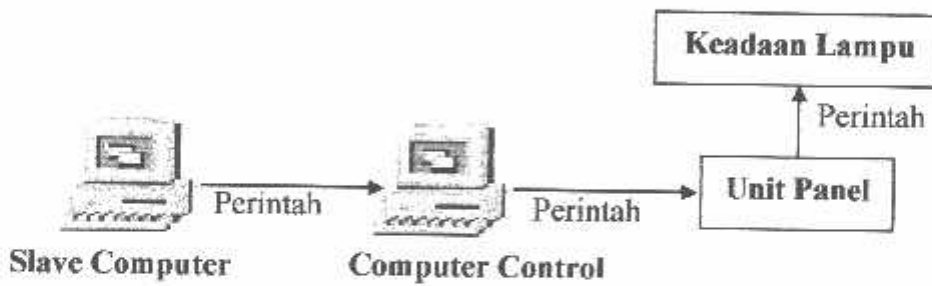
4.3.2. Peralatan

1. *Unit Panel*
2. *Slave Computer.*
3. *Computer Control.*
4. *Ethernet Card.*
5. Kabel UTP (*Unshielded Twister Pair*).

4.3.3. Langkah Pengujian

1. Menghubungkan Kabel Serial antara *Computer Control* dan *Panel*.
 2. Menghubungkan kabel UTP pada *Slave Computer* ke *Computer Control*.
 3. Membuka program Delphi pada dua buah Komputer dan menjalankan program aplikasi.
 4. Pastikan *Computer Control* telah terhubung pada *Unit Panel* dan dapat berkomunikasi.
 5. Seting alamat IP dengan IP lokal 10.0.0.1 untuk *Computer Control* dan 10.0.0.2 untuk *Slave Computer*.
 6. *Connect* kan *server socket* (antara *Computer Control* dan *Slave Computer*). Bila sudah *Connect*, maka
 7. Lakukan Penekanan Tombol
-

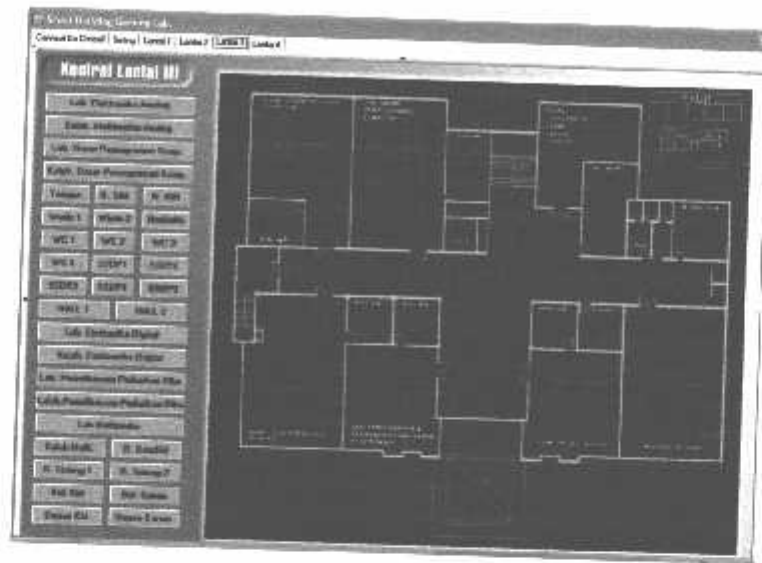
1. Lab. Elektronika Analog, perhatikan perubahan status lampu pada *Slave Computer*, *Computer Control* dan *Smart Building*.
2. Kalab Elektronika Analog, perhatikan perubahan status lampu pada *Slave Computer*, *Computer Control* dan *Smart Building*.
3. Lab Dasar Pemrograman Komputer, perhatikan perubahan status lampu pada *Slave Computer*, *Computer Control* dan *Smart Building*.
4. Lab. Elektronika Digital, perhatikan perubahan status lampu pada *Slave Computer*, *Computer Control* dan *Smart Building*.
5. Kalab. Elektronika Digital, perhatikan perubahan status lampu pada *Slave Computer*, *Computer Control* dan *Smart Building*.



Gambar 4.5 Diagram Blok Perintah

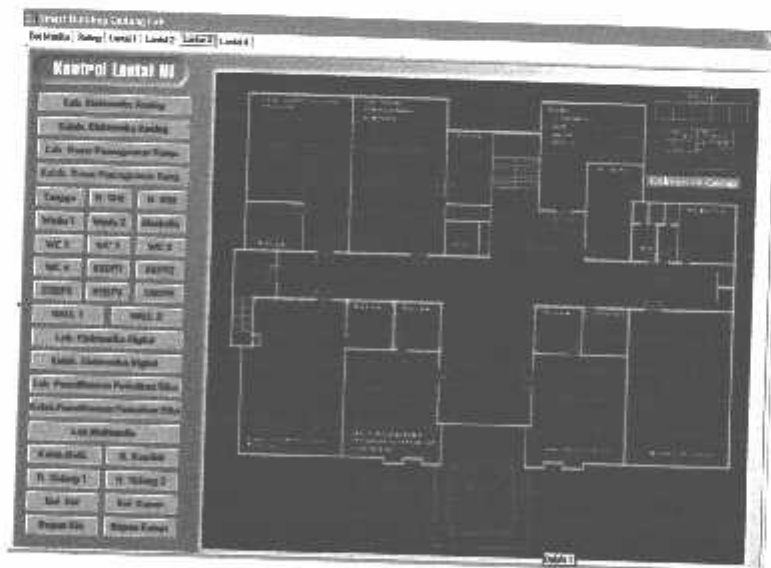
4.3.4. Hasil Pengujian

- Sebelum melakukan penekanan Tombol pada Lab. Elektronika Analog.
 - Tampilan pada *Slave Computer*



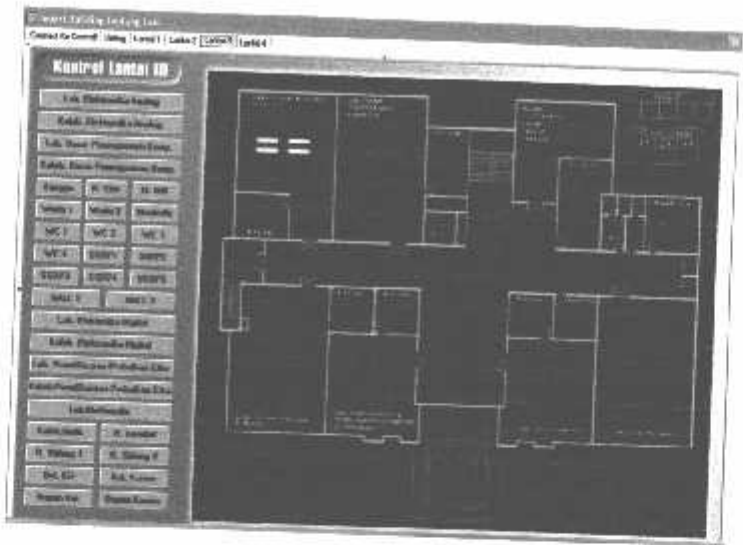
Gambar 4.6 Tampilan pada *Slave Computer* Sebelum melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu

- Tampilan pada *Computer Control*



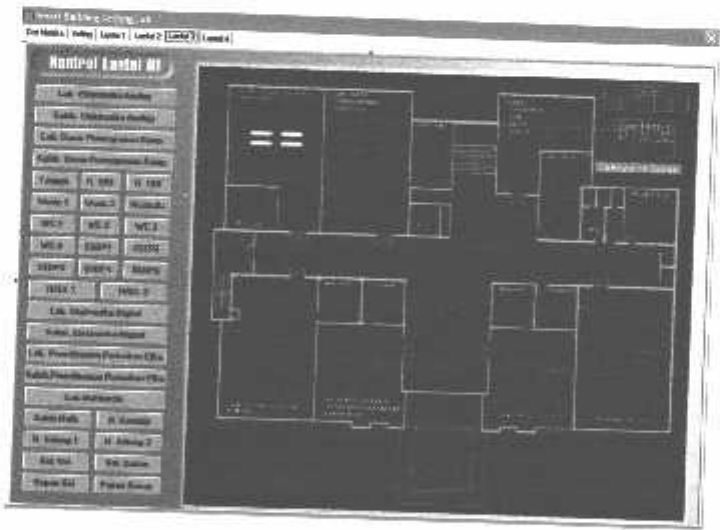
Gambar 4.7 Tampilan pada *Computer Control* Sebelum melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu

- Setelah melakukan penekanan Tombol pada Lab. Elektronika Analog,
 - Tampilan pada *Slave Computer*



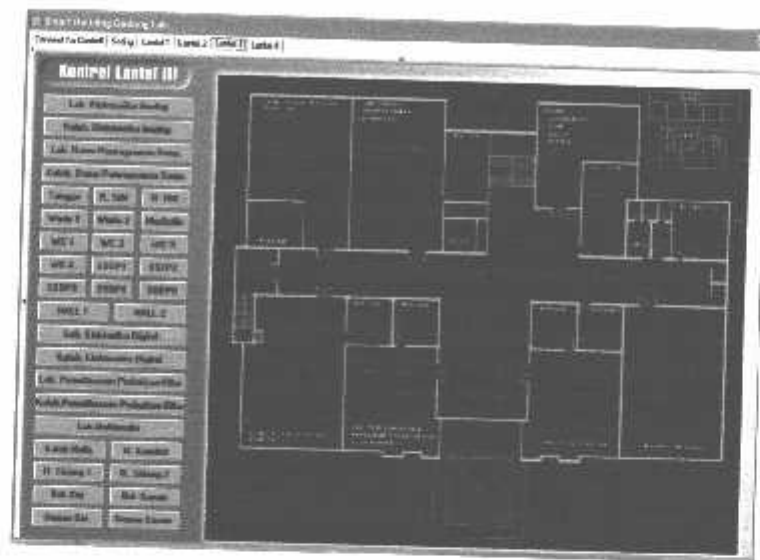
Gambar 4.8 Tampilan pada *Slave Computer* Setelah melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu

- Tampilan pada *Computer Control*



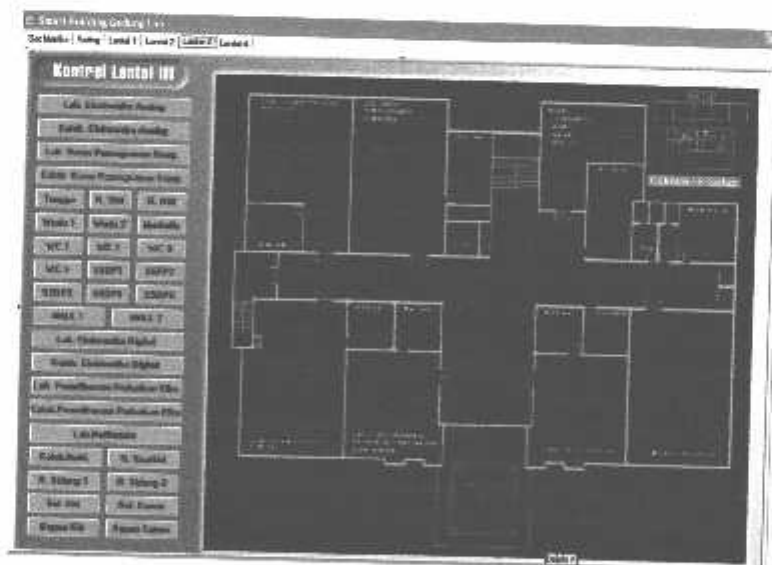
Gambar 4.9 Tampilan pada *Computer Control* Setelah melakukan penekanan Tombol Lab. Elektronika Analog untuk Pengaktifan Lampu

- Setelah melakukan penekanan Tombol pada Lab. Elektronika Analog, untuk pemadaman
 - Tampilan pada *Slave Computer*



Gambar 4.10 Tampilan pada *Slave Computer* Setelah melakukan penekanan Tombol Lab. Elektronika Analog untuk Pemadaman Lampu

- Tampilan pada *Computer Control*



Gambar 4.11 Tampilan pada *Computer Control* Setelah Melakukan Penekanan Tombol Lab. Elektronika Analog untuk Pemadaman Lampu

**Tabel 4.2 Respon Perubahan Status Lampu pada *Computer Control* atas
Permintaan *Slave Computer* Dengan Penekanan
Tombol Lab. Elektronika Analog**

No.	Status Panel	Waktu (s)
1.	OFF ke ON	0,82
2.	ON ke OFF	0,57
3.	OFF ke ON	0,87
4.	ON ke OFF	0,85
5.	OFF ke ON	0,85
6.	ON ke OFF	0,82
7.	OFF ke ON	0,75
8.	ON ke OFF	0,82
9.	OFF ke ON	0,72
10.	ON ke OFF	0,81
11.	OFF ke ON	0,78
12.	ON ke OFF	0,85
13.	OFF ke ON	0,85
14.	ON ke OFF	0,82
15.	OFF ke ON	0,88

Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

$$\text{Waktu Rata-rata} = \frac{\text{Jumlah Waktu}}{\text{Banyaknya Pengujian}}$$

$$= \frac{0,82+0,57+0,87+0,85+0,85+0,82+0,75+0,82+0,72+0,81+0,78+0,85+0,85+0,82+0,88}{15}$$

$$= \frac{12,06}{15} = 0,804s$$

Tabel 4.3 Respon Perubahan Status Panel atas Permintaan *Slave Computer* Dengan Penekanan Tombol Lab. Elektronika Analog

No.	Status Panel	Waktu (s)
1.	OFF ke ON	1,46
2.	ON ke OFF	1,44
3.	OFF ke ON	1,43
4.	ON ke OFF	1,40
5.	OFF ke ON	1,43
6.	ON ke OFF	1,39
7.	OFF ke ON	1,39
8.	ON ke OFF	1,40
9.	OFF ke ON	1,39
10.	ON ke OFF	1,36
11.	OFF ke ON	1,40
12.	ON ke OFF	1,39
13.	OFF ke ON	1,38
14.	ON ke OFF	1,39
15.	OFF ke ON	1,42

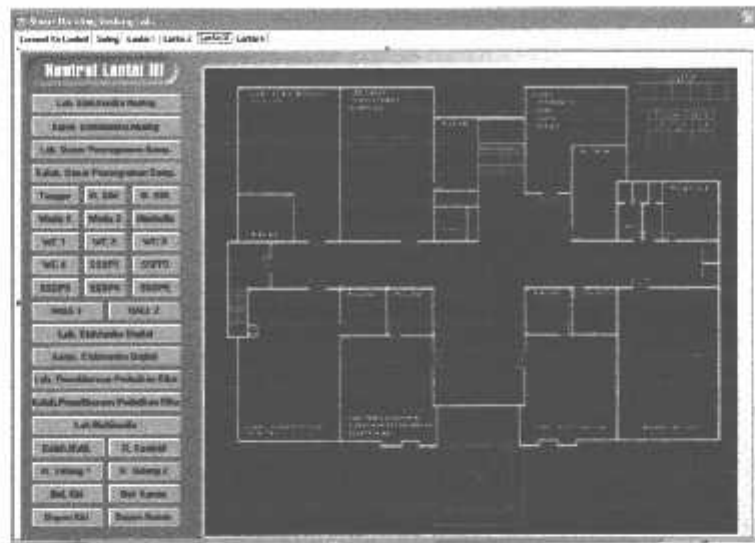
Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

$$\text{Waktu Rata-rata} = \frac{\text{Jumlah Waktu}}{\text{Banyaknya Pengujian}}$$

$$= \frac{1,46+1,44+1,43+1,40+1,43+1,39+1,39+1,40+1,39+1,36+1,40+1,39+1,38+1,39+1,42}{15}$$

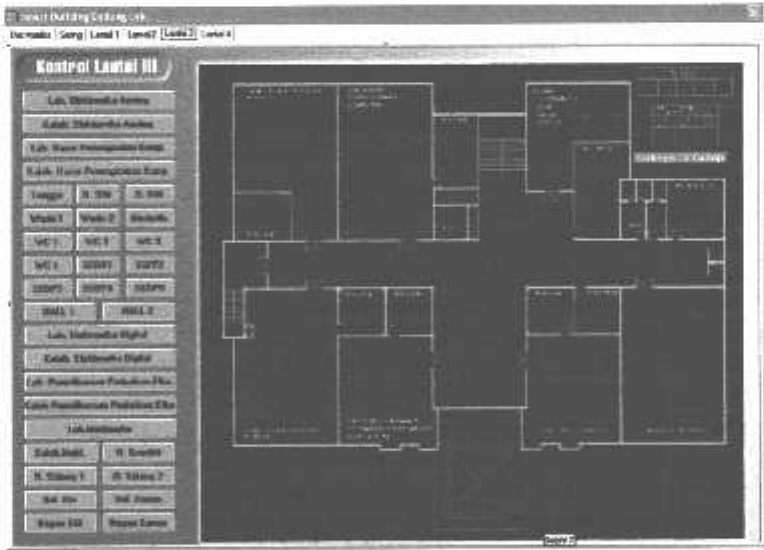
$$= \frac{21,7}{15} = 1,40s$$

- Sebelum melakukan penekanan Tombol Kalab. Elektronika Analog
 - Tampilan pada *Slave Computer*



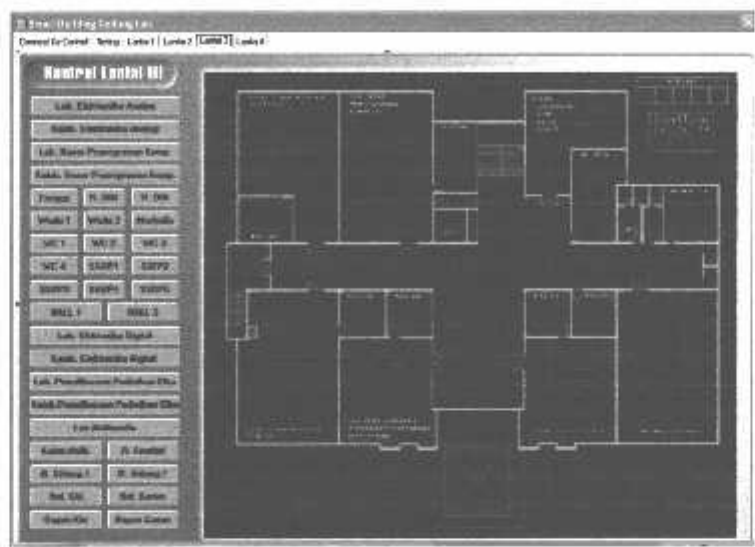
Gambar 4.12 Tampilan pada *Slave Computer* Sebelum melakukan penekanan Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu

- Tampilan pada *Computer Control*



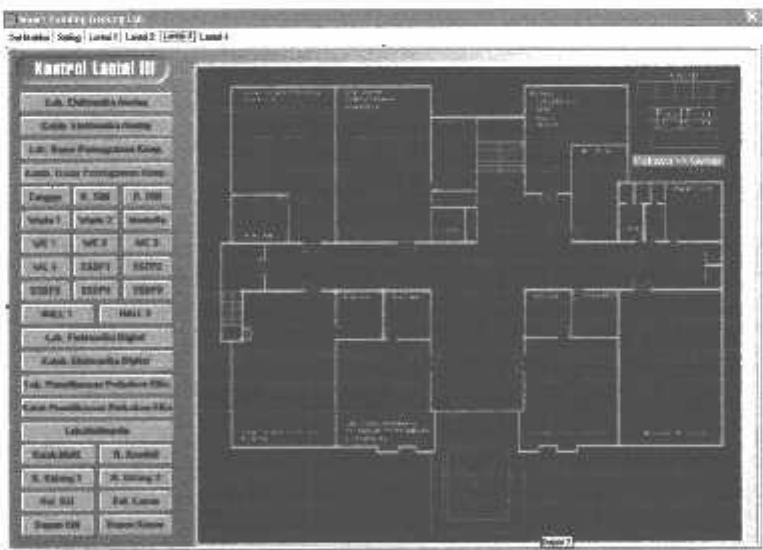
Gambar 4.13 Tampilan pada *Computer Control* Sebelum melakukan penekanan Tombol Kalab. Elektronika Analog untuk Pengaktifan Lampu

- Setelah melakukan penekanan Tombol pemadaman pada Kalab. Elektronika Analog,
 - Tampilan pada *Slave Computer*



Gambar 4.16 Tampilan pada *Slave Computer* Setelah melakukan penekanan Tombol Kalab. Elektronika Analog untuk Pemadaman Lampu

- Tampilan pada *Computer Control*



Gambar 4.17 Tampilan pada *Computer Control* Setelah melakukan penekanan Tombol Kalab. Elektronika Analog untuk Pemadaman Lampu

**Tabel 4.4 Respon Perubahan Status Lampu pada *Computer Control* atas
Permintaan *Slave Computer*
Dengan Penekanan Tombol Kalah. Elektronika Analog**

No.	Status Panel	Waktu (s)
1.	OFF ke ON	0,88
2.	ON ke OFF	0,84
3.	OFF ke ON	0,88
4.	ON ke OFF	0,87
5.	OFF ke ON	0,88
6.	ON ke OFF	0,86
7.	OFF ke ON	0,82
8.	ON ke OFF	0,82
9.	OFF ke ON	0,87
10.	ON ke OFF	0,88
11.	OFF ke ON	0,83
12.	ON ke OFF	0,86
13.	OFF ke ON	0,83
14.	ON ke OFF	0,90
15.	OFF ke ON	0,88

Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

$$\text{Waktu Rata-rata} = \frac{\text{Jumlah Waktu}}{\text{Banyaknya Pengujian}}$$

$$= \frac{0,88+0,84+0,88+0,87+0,88+0,86+0,82+0,82+0,87+0,88+0,83+0,86+0,83+0,90+0,88}{15}$$

$$= \frac{12,9}{15} = 0,86s$$

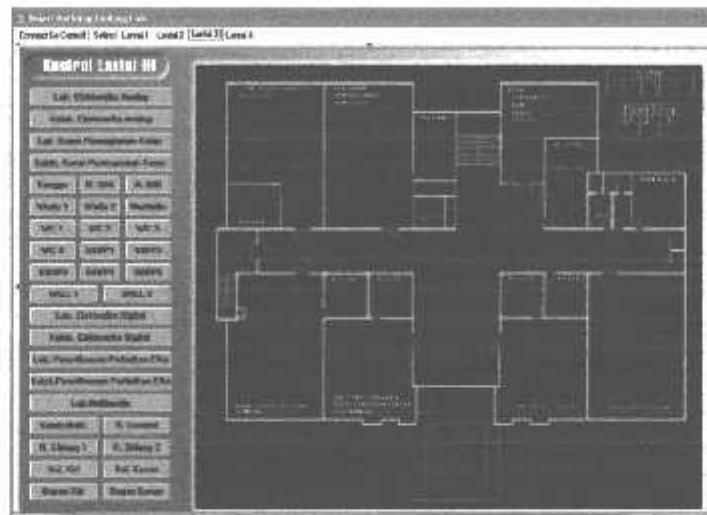
Tabel 4.5 Respon Perubahan Status Panel atas Permintaan *Slave Computer*
Dengan Penekanan Tombol Kalab. Elektronika Analog

No.	Status Panel	Waktu (s)
1.	OFF ke ON	1,40
2.	ON ke OFF	1,40
3.	OFF ke ON	1,42
4.	ON ke OFF	1,44
5.	OFF ke ON	1,37
6.	ON ke OFF	1,48
7.	OFF ke ON	1,52
8.	ON ke OFF	1,36
9.	OFF ke ON	1,38
10.	ON ke OFF	1,42
11.	OFF ke ON	1,75
12.	ON ke OFF	1,45
13.	OFF ke ON	1,58
14.	ON ke OFF	1,50
15.	OFF ke ON	1,46

Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

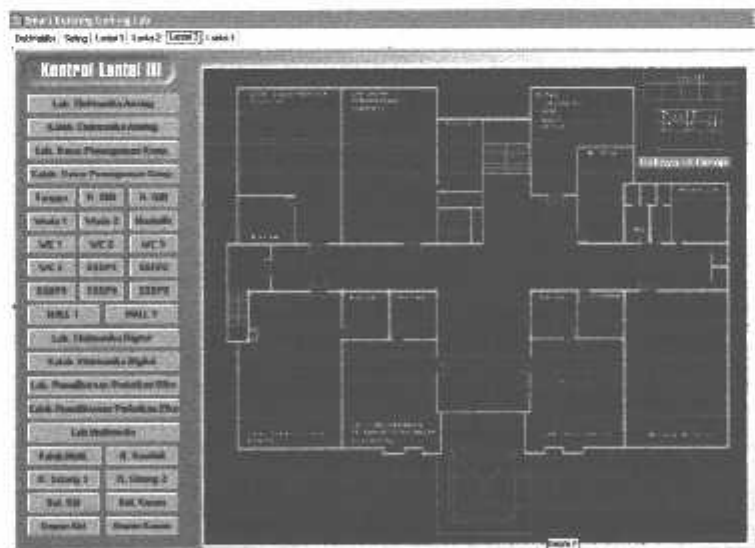
$$\begin{aligned}
 \text{Waktu Rata-rata} &= \frac{\text{Jumlah Waktu}}{\text{Banyaknya Pengujian}} \\
 &= \frac{1,40+1,40+1,42+1,44+1,37+1,48+1,52+1,36+1,38+1,42+1,75+1,45+1,58+1,50+1,46}{15} \\
 &= \frac{21,93}{15} = 1,462s
 \end{aligned}$$

- Setelah melakukan penekanan Tombol pemadaman pada Lab. Elektronika Digital
- Tampilan pada *Slave Computer*



Gambar 4.22 Tampilan pada *Slave Computer* Setelah melakukan penekanan Tombol Lab. Elektronika Digital untuk Pemadaman Lampu

- Tampilan pada *Computer Control*



Gambar 4.23 Tampilan pada *Computer Control* Setelah melakukan penekanan Tombol Lab. Elektronika Digital untuk Pemadaman Lampu

**Tabel 4.6 Respon Perubahan Status Lampu pada *Computer Control* atas
Permintaan *Slave Computer*
Dengan Penekanan Tombol Lab. Elektronika Digital**

Penekanan	Status Panel	Waktu (s)
1.	OFF ke ON	0,82
2.	ON ke OFF	0,81
3.	OFF ke ON	0,92
4.	ON ke OFF	0,87
5.	OFF ke ON	0,84
6.	ON ke OFF	0,94
7.	OFF ke ON	0,96
8.	ON ke OFF	0,82
9.	OFF ke ON	0,79
10.	ON ke OFF	0,85
11.	OFF ke ON	0,88
12.	ON ke OFF	0,82
13.	OFF ke ON	0,91
14.	ON ke OFF	0,84
15.	OFF ke ON	0,85

Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

$$\text{Waktu Rata-rata} = \frac{\text{Jumlah Waktu}}{\text{Banyaknya Pengujian}}$$

$$= \frac{0,82+0,81+0,92+0,87+0,84+0,94+0,96+0,82+0,79+0,85+0,88+0,82+0,91+0,84+0,85}{15}$$

$$= \frac{12,92}{15} = 0,86s$$

Tabel 4.7 Respon Perubahan Status Panel atas Permintaan *Slave Computer*
Dengan Penekanan Tombol Lab. Elektronika Digital

Penekanan	Status Panel	Waktu (s)
1.	OFF ke ON	1,58
2.	ON ke OFF	1,42
3.	OFF ke ON	1,51
4.	ON ke OFF	1,51
5.	OFF ke ON	1,56
6.	ON ke OFF	1,50
7.	OFF ke ON	1,53
8.	ON ke OFF	1,42
9.	OFF ke ON	1,50
10.	ON ke OFF	1,45
11.	OFF ke ON	1,45
12.	ON ke OFF	1,40
13.	OFF ke ON	1,46
14.	ON ke OFF	1,46
15.	OFF ke ON	1,45

Waktu Rata-rata dari pengujian sebanyak 15 kali didapatkan:

$$\text{Waktu Rata-rata} = \frac{\text{Jumlah_Waktu}}{\text{Banyaknya_Pengujian}}$$

$$= \frac{1,58+1,42+1,51+1,51+1,56+1,50+1,53+1,42+1,50+1,45+1,45+1,40+1,46+1,46+1,45}{15}$$

$$= \frac{22,18}{15} = 1,479s$$



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari Pengujian yang dilakukan dengan alat ini, maka dapat diambil kesimpulan bahwa :

1. Dengan menggunakan komponen *Socket*, data yang diterima sisi penerima sesuai dengan data asli yang dikirim sisi pengirim.
2. Di ambil dari perhitungan rata-rata, respon perubahan *Panel* atas permintaan dari *Slave Computer* Tidak lebih dari 1,5 detik.
3. Jika lebih dari 1,5 detik *Panel* tidak melakukan peruban status, berarti data yang dikirimkan tidak diterima oleh *Panel*.

5.2. Saran

1. Untuk Mengetahui apakah *Panel* benar-benar melakukan perubahan statu atas perintah *Slave Computer*, Tunggu dalam waktu maksimal 19,5detik (proses sekali siklus *Slave Computer* melakukan pengecekan data setiap *Panel* melalui *Computer Control*).



DAFTAR PUSTAKA

- [1]. Tjatur, P. Johanes, 2001, Pelatihan Mikrokontroler MCS-51 Programing and Interfacing, Malang.
 - [2]. Nalwan, Andi Paulus, 2003, Teknik Antarmuka dan Pemrograman, Jakarta
 - [3]. Suthanto , 2001
 - [4]. www.atmel.com, (Data sheet EEPROM AT24C01A/02/04/08A/16A)
 - [5]. Dallas semiconductor, tahun1998.
 - [6]. ARC Electronics, 2000.
 - [7]. www.bb-elec.com, 1997.
 - [8]. www.maxim-ic.com,
 - [9]. R.H Smith, 1999.
 - [10]. Rachmad Setiawan, 2005.
 - [12]. *Menjadi Administrator Jaringan Komputer*, Penerbit Andi, Yogyakarta tahun 2005.
 - [13]. Antony Pranata, *Pemrograman Borland Delphi 7*, Edisi 4, Penerbit: Andi, Yogyakarta. 2002.
 - [14]. Inge Martina, Ir, *36 Jam Belajar Komputer Visual Borland Delphi 7*, Elex Media Kompatindo, Jakarta, tahun 2004.
 - [15]. Inge Martina, Ir, *36 Jam Belajar Komputer Pemrograman Internet dengan Delphi*, Elcx Media Kompatindo, Jakarta, tahun 2002.
 - [16]. http://id.wikipedia.org/wiki/Port_TCP_dan_UDP"
 - [17]. Smart Building Team ITN Malang, tahun 2005.
-





LAMPIRAN



FORMULIR BIMBINGAN SKRIPSI

Nama : EKO APRIYANTO
Nim : 03 17 109
Masa Bimbingan : 05 JUNI 2007 s/d 05 DESEMBER 2007
Judul Skripsi : PERENCANAAN PEMBUATAN SLAVE COMPUTER
CONTROL UNTUK PENGOPERASIAN SEMART
BUILDING JURUSAN TEKNIK ELEKTRO ITN MALANG

NO.	Tanggal	Uraian	Paraf Pembimbing
1.	30/07'07	Bab I, II, III Revisi	
2.	3/08'07	Bab III Revisi FlowChart	
3.	07/08'07	Bab IV Revisi	
4.	14/08'07	Bab V Revisi kesimpulan	
5.	21/08'07	Bab V Saran ditambah	
6.	23/08'07	Bab III Acc	
7.	24/08'07	Bab IV, V Acc	
8.	31/8 2007	Acc lengkap	
9.			
10.			

Malang,
Dosen Pembimbing I

Joseph Dedy Irawan, ST., MT.
NIP. 132315178

Form S-4



FORMULIR BIMBINGAN SKRIPSI

Nama : EKO APRIYANTO
Nim : 03 17 109
Masa Bimbingan : 05 JUNI 2007 s/d 05 DESEMBER 2007
Judul Skripsi : PERENCANAAN PEMBUATAN SLAVE COMPUTER
CONTROL UNTUK PENGOPERASIAN SEMART
BUILDING JURUSAN TEKNIK ELEKTRO ITN MALANG

NO.	Tanggal	Uraian	Paraf Pembimbing
1.	21/07/08	Revisi Bab I, Bab II	
2.	28/07/08	Revisi Bab III	
3.	30/07/08	Revisi Bab I, Bab II + Revisi Bab III	
4.	31/07/08	Revisi seluruh Bab (Laporan Harian)	
5.			
6.			
7.			
8.			
9.			
10.			

Malang, 31-08-2007
Dosen Pembimbing II

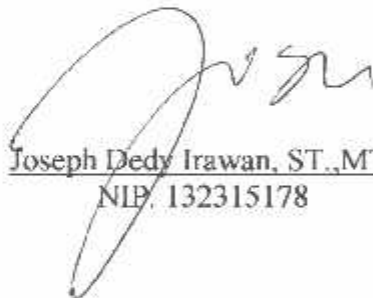
I Komang Somawirata, ST.,MT.
NIP.P. 1030100365



LEMBAR BIMBINGAN SKRIPSI

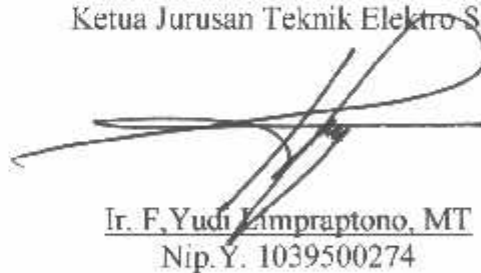
Nama : Eko Apriyanto
NIM : 03 17 109
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Masa Bimbingan : 05 Juni 2007 s/d 05 Desember 2007
Judul Skripsi : *Perencanaan Pembuatan Slave Computer Control Untuk Pengoperasian Smart Building Jurusan Teknik Elektro ITN Malang*
Dosen Pembimbing : Joseph Dedy Irawan, ST.,MT.
Telah Dievaluasi : 93 (Sembilan Puluh Tiga)

Diperiksa dan Disetujui,
Dosen Pembimbing I



Joseph Dedy Irawan, ST.,MT.
NIP. 132315178

Mengetahui,
Ketua Jurusan Teknik Elektro S-1



Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274



LEMBAR BIMBINGAN SKRIPSI

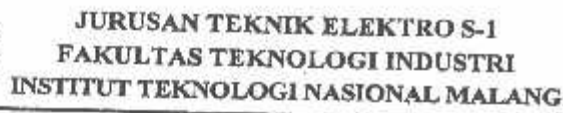
Nama : Eko Apriyanto
NIM : 03 17 109
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Masa Bimbingan : 05 Juni 2007 s/d 05 Desember 2007
Judul Skripsi : Perencanaan Pembuatan *Slave Computer Control* Untuk
Pengoperasian *Semart Building* Jurusan Teknik Elektro
ITN Malang
Dosen Pembimbing : I Komang Somawirata, ST.,MT
Telah Dievaluasi : 92 (Sembilan Puluh Dua)

Diperiksa dan Disetujui,
Dosen Pembimbing II

I Komang Somawirata, ST.,MT.
NIP.P. 1030100365

Mengetahui,
Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
Nip. Y. 1039500274



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi, perlu adanya perbaikan skripsi untuk mahasiswa :

Nama	:	Eko Apriyanto						
NIM	:	0317109						
Fakultas	:	Teknologi Industri						
Jurusan	:	Teknik Elektro S-1						
Konsentrasi	:	<table border="1"> <tr> <td>1.</td> <td>Teknik Energi Listrik *)</td> </tr> <tr> <td>2.</td> <td>Teknik Elektronika *)</td> </tr> <tr> <td>3.</td> <td>Tenik Komputer dan Informatika *)</td> </tr> </table>	1.	Teknik Energi Listrik *)	2.	Teknik Elektronika *)	3.	Tenik Komputer dan Informatika *)
1.	Teknik Energi Listrik *)							
2.	Teknik Elektronika *)							
3.	Tenik Komputer dan Informatika *)							

Perbaikan meliputi :

1. Sempurnakan software agar pada proses pengisian instruksi tidak gagal, serta proses pengisian instruksi tidak perlu dilakukan lagi secara manual, tapi secara otomatis. (Spt pada lampiran 3.)

Catatan :

*) Coret yang tidak diperlukan.

Malang, _____

Dosen Penguji

(W. H. G. Smith)



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi, perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Eko Apriyanto
NIM : 03 17 109
Masa Bimbingan : 05 Juni 2007 s/d 05 Desember 2007
Judul Skripsi : Perencanaan Pembuatan *Slave Computer Control* untuk Pengoperasian *Smart Building* Jurusan Teknik Elektro ITN Malang

Tanggal	Uraian	Paraf
03 September 2007	Sempurnakan Software agar pada proses pengiriman instruksi kegagalan, maka pengiriman instruksi tidak perlu dilakukan lagi secara manual, tapi secara otomatis.	

Mengetahui,

Dosen Pembimbing I

Joseph Dedy Irawan, ST.,MT.
NIP. 132315178

Dosen Pembimbing II

I Komang Somawirata, ST.,MT.
NIP. 1030100365

Dosen Penguji,

Penguji I

Ir. Widodo Puji Mulyanto, MT.
NIP.Y. 1028700171

Penguji II

Dr. Cahyo Crysdian, Msc.
NIP. 1030400412

Features

- Compatible with MCS-51[®] Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.



8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

Rev. 2487A-10/01



Pin Configurations

PDIP

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
(MOSI) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

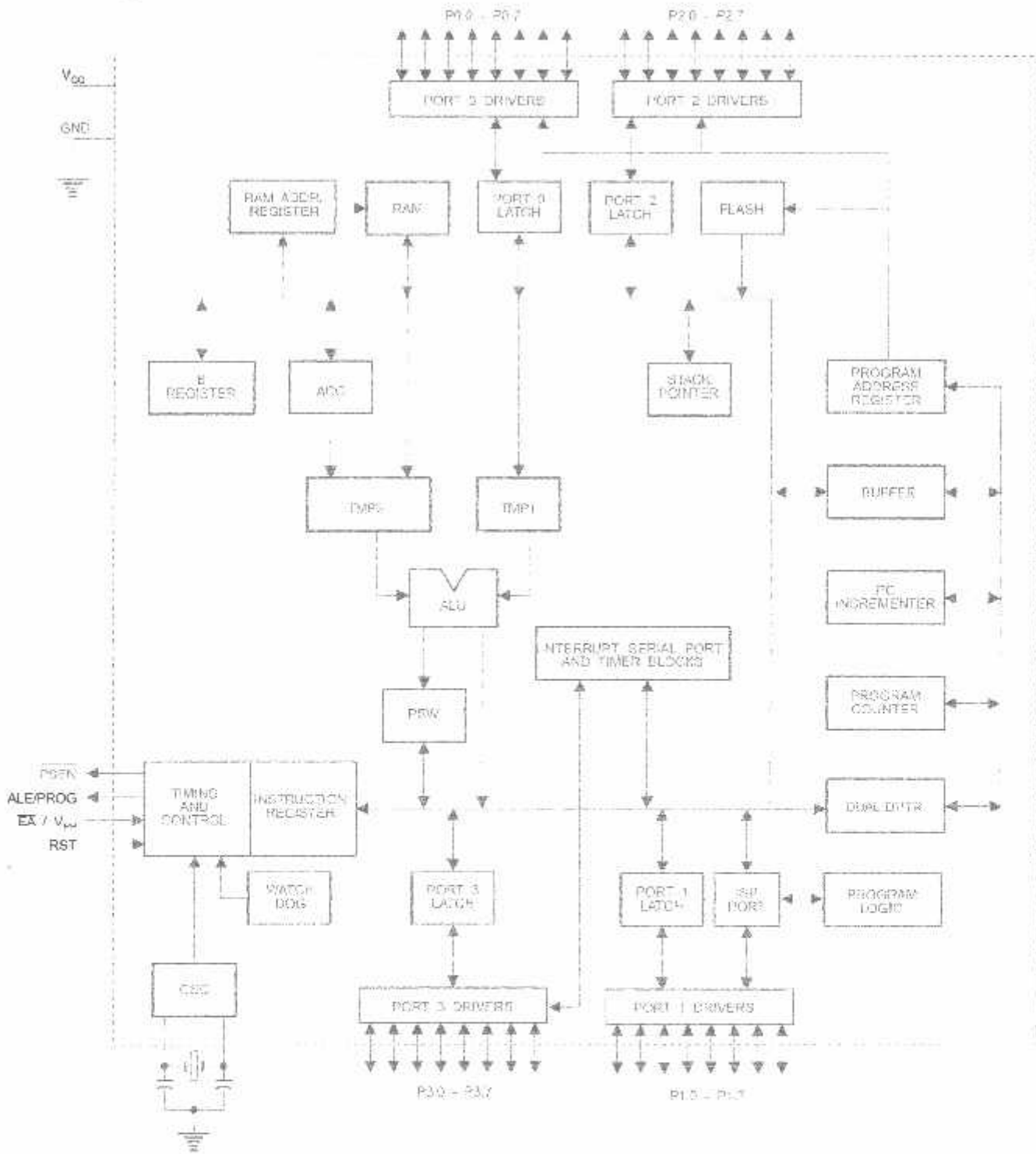
PLCC

P1.4	6	P1.3	5	P1.2	4	P1.1	3	P1.0	2	1	NC	44	VCC	43	P0.0 (AD0)	42	P0.1 (AD1)	41	P0.2 (AD2)	40	P0.3 (AD3)	39	P0.4 (AD4)	38	P0.5 (AD5)	37	P0.6 (AD6)	36	P0.7 (AD7)	35	EA/VPP	34	NC	33	ALE/PROG	32	PSEN	31	P2.7 (A15)	30	P2.6 (A14)	29	P2.5 (A13)	28	P2.4 (A12)	27	P2.3 (A11)	26	P2.2 (A10)	25	P2.1 (A9)	24	P2.0 (A8)	23	GND	22	NC	21	XTAL1	20	XTAL2	19	(RD) P3.7	18	(WR) P3.6	17	(T1) P3.5	16	(T0) P3.4	15	(INT1) P3.3	14	(INT0) P3.2	13	(TXD) P3.1	12	NC	11	(RXD) P3.0	10	RST	9	(SCK) P1.7	8	(MISO) P1.6	7	(MOSI) P1.5	6	P1.4	5	P1.3	4	P1.2	3	P1.1	2	P1.0	1
------	---	------	---	------	---	------	---	------	---	---	----	----	-----	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	--------	----	----	----	----------	----	------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	-----------	----	-----------	----	-----	----	----	----	-------	----	-------	----	-----------	----	-----------	----	-----------	----	-----------	----	-------------	----	-------------	----	------------	----	----	----	------------	----	-----	---	------------	---	-------------	---	-------------	---	------	---	------	---	------	---	------	---	------	---

TQFP

44	P1.4	43	P1.3	42	P1.2	41	P1.1	40	P1.0	39	NC	38	VCC	37	P0.0 (AD0)	36	P0.1 (AD1)	35	P0.2 (AD2)	34	P0.3 (AD3)	33	P0.4 (AD4)	32	P0.5 (AD5)	31	P0.6 (AD6)	30	P0.7 (AD7)	29	EA/VPP	28	NC	27	ALE/PROG	26	PSEN	25	P2.7 (A15)	24	P2.6 (A14)	23	P2.5 (A13)	22	P2.4 (A12)	21	P2.3 (A11)	20	P2.2 (A10)	19	P2.1 (A9)	18	P2.0 (A8)	17	GND	16	GND	15	XTAL1	14	XTAL2	13	(RD) P3.7	12	(WR) P3.6	11	(T1) P3.5	10	(T0) P3.4	9	(INT1) P3.3	8	(INT0) P3.2	7	(TXD) P3.1	6	NC	5	(RXD) P3.0	4	RST	3	(SCK) P1.7	2	(MISO) P1.6	1	(MOSI) P1.5
----	------	----	------	----	------	----	------	----	------	----	----	----	-----	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	--------	----	----	----	----------	----	------	----	------------	----	------------	----	------------	----	------------	----	------------	----	------------	----	-----------	----	-----------	----	-----	----	-----	----	-------	----	-------	----	-----------	----	-----------	----	-----------	----	-----------	---	-------------	---	-------------	---	------------	---	----	---	------------	---	-----	---	------------	---	-------------	---	-------------

Block Diagram



Pin Description

V_{CC} Supply voltage.

V_{IND} Ground.

Port 0 Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

Port 1 Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2 Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3 Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

RESET Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/PROG Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$ Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA/VP}}$ External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

$\overline{\text{TAL1}}$ Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

$\overline{\text{TAL2}}$ Output from the inverting oscillator amplifier





Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S51 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000
								87H

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the five interrupt sources in the IP register.

Table 2. AUXR: Auxiliary Register

AUXR	Address = 8EH				Reset Value = XXX00XX0B			
Not Bit Addressable								
Bit	7	6	5	WDIDLE	DISRTO	2	1	DISALE
— Reserved for future expansion								
DISALE Disable/Enable ALE								
DISALE Operating Mode								
0 ALE is emitted at a constant rate of 1/6 the oscillator frequency								
1 ALE is active only during a MOVX or MOVC instruction								
DISRTO Disable/Enable Reset out								
DISRTO								
0 Reset pin is driven High after WDT times out								
1 Reset pin is input only								
WDIDLE Disable/Enable WDT in IDLE mode								
WDIDLE								
0 WDT continues to count in IDLE mode								
1 WDT halts counting in IDLE mode								

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.



Power Off Flag: The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by reset.

Table 3. AUXR1: Auxiliary Register 1

AUXR1							
Address = A2H							
Reset Value = XXXXXX0B							
Not Bit Addressable							
Bit	7	6	5	4	3	2	DPS
	–	–	–	–	–	–	0
– Reserved for future expansion							
DPS Data Pointer Register Select							
DPS							
0 Selects DPTR Registers DP0L, DP0H							
1 Selects DPTR Registers DP1L, DP1H							

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the \overline{EA} pin is connected to GND, all program fetches are directed to external memory. On the AT89S51, if \overline{EA} is connected to V_{CC} , program fetches to addresses 0000H through FFFH are directed to internal memory and fetches to addresses 1000H through FFFFH are directed to external memory.

Data Memory

The AT89S51 implements 128 bytes of on-chip RAM. The 128 bytes are accessible via direct and indirect addressing modes. Stack operations are examples of indirect addressing, so the 128 bytes of data RAM are available as stack space.

Watchdog Timer One-time Enabled with Reset-out

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 98xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it

should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt, which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S51 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S51 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 4 shows that bit position IE.6 is unimplemented. In the AT89S51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

The Timer 0 and Timer 1 flags, TFC and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle.

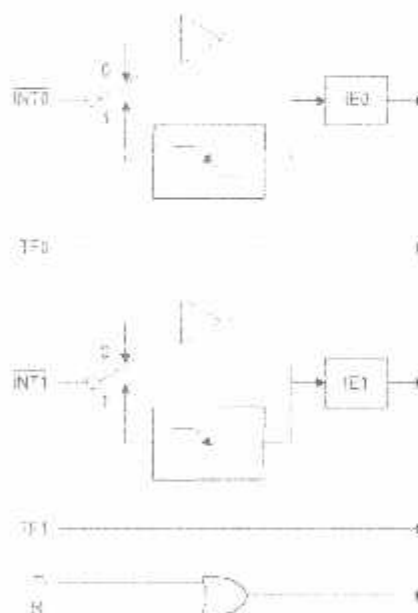
Table 4. Interrupt Enable (IE) Register

(MSB)				(LSB)			
EA	—	—	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved
—	IE.5	Reserved
ES	IE.4	Serial Port interrupt enable bit
ET1	IE.3	Timer 1 interrupt enable bit
EX1	IE.2	External interrupt 1 enable bit
ET0	IE.1	Timer 0 interrupt enable bit
EX0	IE.0	External interrupt 0 enable bit

User software should never write 1s to reserved bits, because they may be used in future AT89 products.

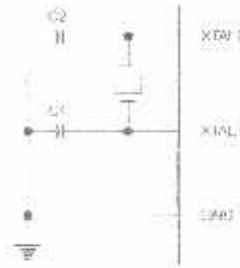
Figure 1. Interrupt Sources



Oscillator Characteristics

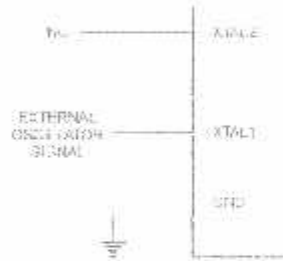
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



Note: C1, C2 = 30 pF \pm 10 pF for Crystals = 40 pF \pm 10 pF for Ceramic Resonators

Figure 3. External Clock Drive Configuration



Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special function registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by activation of an enabled external interrupt into INT0 or INT1. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.



Table 5. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Program
Memory Lock
bits

The AT89S51 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 6. Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features
2	P	U	U	MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the Flash memory is disabled
3	P	P	U	Same as mode 2, but verify is also disabled
4	P	P	P	Same as mode 3, but external execution is also disabled

When lock bit 1 is programmed, the logic level at the EA pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of EA must agree with the current logic level at that pin in order for the device to function properly.

Programming
the Flash –
Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S51, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise EA/V_{PP} to 12V.
5. Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 μs. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89S51 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.0 is pulled low after ALE goes high during programming to indicate **BUSY**. P3.0 is pulled high again when programming is done to indicate **READY**.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 51H indicates 89S51
(200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/PROG low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

1. **Power-up sequence:**
Apply power between VCC and GND pins.
Set RST pin to "H".
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.



Power-off sequence (if needed):
 Set XTAL1 to "L" (if a crystal is not used).
 Set RST to "L".
 Turn V_{CC} power off.

Data Polling: The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

**Serial
Programming
Instruction Set**






The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 8 on page 18.

**Programming
Interface –
Parallel Mode**

Every code byte in the Flash array can be programmed by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 7. Flash Programming Modes

Mode	V _{CC}	RST	PSEN	ALE/ PROG	EA/ V _{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.3-0	P1.7-0
												Address	
Write Code Data	5V	H	L		12V	L	H	H	H	H	D _{IN}	A11-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D _{OUT}	A11-8	A7-0
Write Lock Bit 1	5V	H	L		12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L		12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L		12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L		12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	51H	0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	0010	00H

- Notes:
1. Each PROG pulse is 200 ns - 500 ns for Chip Erase.
 2. Each PROG pulse is 200 ns - 500 ns for Write Code Data.
 3. Each PROG pulse is 200 ns - 500 ns for Write Lock Bits.
 4. RDY/BSY signal is output on P3.0 during programming.
 5. X = don't care.

Flash Programming and Verification Characteristics (Parallel Mode)

- 20°C to 30°C, V_{CC} = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
P	Programming Supply Voltage	11.5	12.5	V
	Programming Supply Current		10	mA
	V _{CC} Supply Current		30	mA
CLCL	Oscillator Frequency	3	33	MHz
GL	Address Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
AX	Address Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
GL	Data Setup to $\overline{\text{PROG}}$ Low	48t _{CLCL}		
DX	Data Hold After $\overline{\text{PROG}}$	48t _{CLCL}		
SH	P2.7 (ENABLE) High to V _{PP}	48t _{CLCL}		
GL	V _{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
SI	V _{PP} Hold After $\overline{\text{PROG}}$	10		μs
GH	$\overline{\text{PROG}}$ Width	0.2	1	μs
QV	Address to Data Valid		48t _{CLCL}	
QV	ENABLE Low to Data Valid		48t _{CLCL}	
QZ	Data Float After ENABLE	0	48t _{CLCL}	
BL	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
W	Byte Write Cycle Time		50	μs

Figure 6. Flash Programming and Verification Waveforms – Parallel Mode

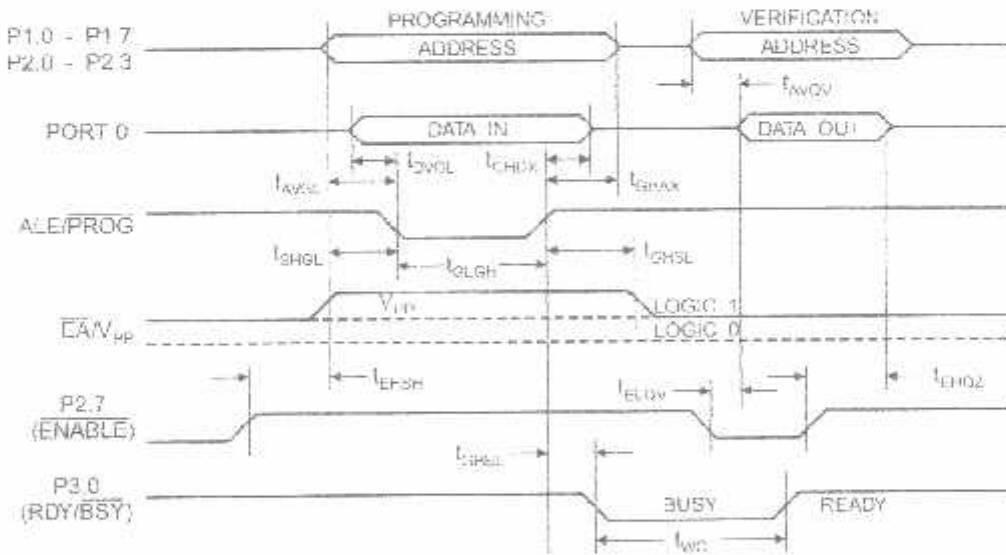
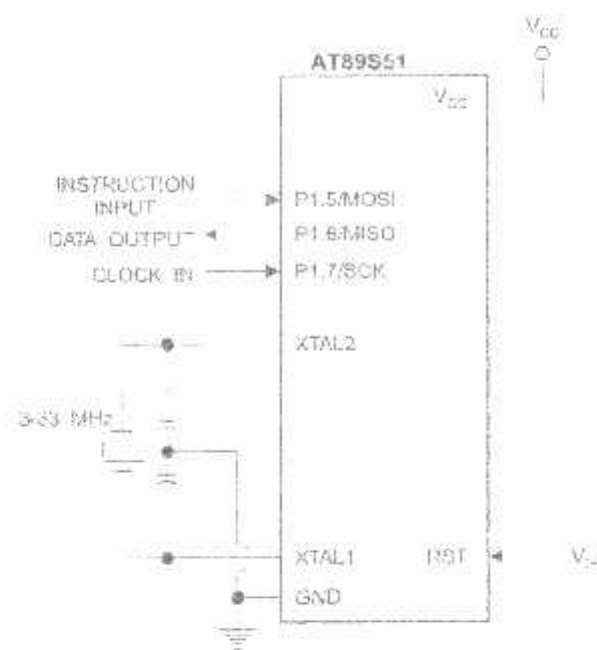
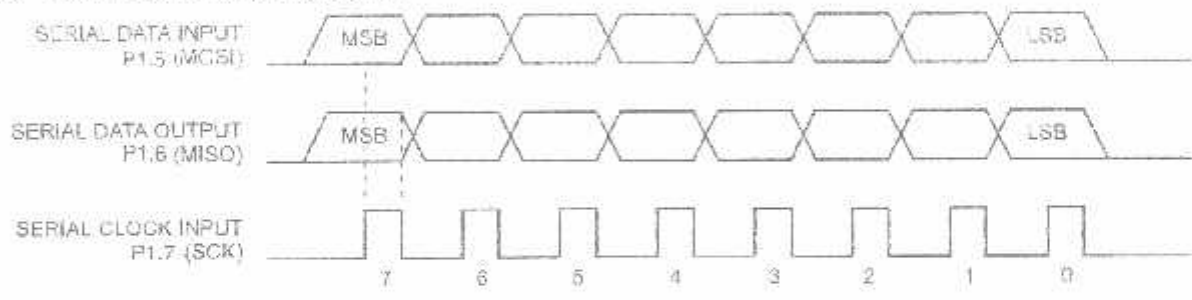


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms





bsolute Maximum Ratings*

erating Temperature	-55°C to +125°C
orage Temperature	-65°C to +150°C
oltage on Any Pin h Respect to Ground	-1.0V to +7.0V
imum Operating Voltage	6.6V
Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Characteristics

values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

mbol	Parameter	Condition	Min	Max	Units
	Input Low Voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC}-0.1$	V
\overline{t}	Input Low Voltage (\overline{EA})		-0.5	$0.2 V_{CC}-0.3$	V
t	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC}+0.9$	$V_{CC}+0.5$	V
t	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC}+0.5$	V
I_L	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
I_{L1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, \overline{PSEN})	$I_{OL} = 3.2 \text{ mA}$		0.45	V
I_{OH}	Output High Voltage (Ports 1,2,3, ALE, \overline{PSEN})	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
I_{OHT}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45V$		-50	μA
I_{L}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2V, V_{CC} = 5V \pm 10\%$		-650	μA
I_I	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RST	Reset Pulldown Resistor		50	300	K Ω
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T _A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 5.5V$		50	μA

- Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

AT89S51

Characteristics

Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other ports = 80 pF.

Internal Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
f _{OSC}	Oscillator Frequency			0	33	MHz
t _{PL}	ALE Pulse Width	127		2t _{CLCL} -40		ns
t _{VL}	Address Valid to ALE Low	43		t _{CLCL} -25		ns
t _{AX}	Address Hold After ALE Low	48		t _{CLCL} -25		ns
t _{IV}	ALE Low to Valid Instruction In		233		4t _{CLCL} -65	ns
t _{PL}	ALE Low to PSEN Low	43		t _{CLCL} -25		ns
t _{PH}	PSEN Pulse Width	205		3t _{CLCL} -45		ns
t _{IV}	PSEN Low to Valid Instruction In		145		3t _{CLCL} -60	ns
t _{IX}	Input Instruction Hold After PSEN	0		0		ns
t _{IZ}	Input Instruction Float After PSEN		59		t _{CLCL} -25	ns
t _{XAV}	PSEN to Address Valid	75		t _{CLCL} -8		ns
t _{VV}	Address to Valid Instruction In		312		5t _{CLCL} -80	ns
t _{LAZ}	PSEN Low to Address Float		10		10	ns
t _{RDH}	RD Pulse Width	400		6t _{CLCL} -100		ns
t _{WRH}	WR Pulse Width	400		6t _{CLCL} -100		ns
t _{LDV}	RD Low to Valid Data In		252		5t _{CLCL} -90	ns
t _{HDX}	Data Hold After RD	0		0		ns
t _{HDZ}	Data Float After RD		97		2t _{CLCL} -28	ns
t _{LDV}	ALE Low to Valid Data In		517		8t _{CLCL} -150	ns
t _{VDDV}	Address to Valid Data In		585		9t _{CLCL} -165	ns
t _{WL}	ALE Low to RD or WR Low	200	300	3t _{CLCL} -50	3t _{CLCL} +50	ns
t _{AVWL}	Address to RD or WR Low	203		4t _{CLCL} -75		ns
t _{QVWX}	Data Valid to WR Transition	23		t _{CLCL} -30		ns
t _{QVWH}	Data Valid to WR High	433		7t _{CLCL} -130		ns
t _{WHQX}	Data Hold After WR	33		t _{CLCL} -25		ns
t _{RLAZ}	RD Low to Address Float		0		0	ns
t _{WHLH}	RD or WR High to ALE High	43	123	t _{CLCL} -25	t _{CLCL} +25	ns



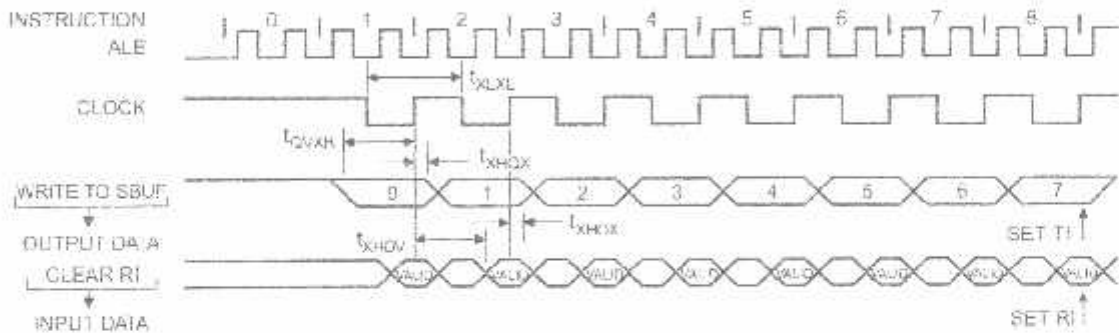
[illegible][illegible]

Serial Port Timing: Shift Register Mode Test Conditions

Values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = 80 pF .

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{CLK}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
$t_{D/XH}$	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
$t_{D/X}$	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-80$		ns
$t_{D/X}$	Input Data Hold After Clock Rising Edge	0		0		ns
$t_{D/V}$	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms



C Testing Input/Output Waveforms⁽¹⁾



Note: 1. AC Inputs during testing are driven at $V_{CC} = 0.5V$ for a logic 1 and $0.45V$ for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Load Waveforms⁽¹⁾



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AC	44A	Commercial (0° C to 70° C)
		AT89S51-24JC	44J	
		AT89S51-24PC	40P6	
		AT89S51-24AI	44A	Industrial (-40° C to 85° C)
		AT89S51-24JI	44J	
		AT89S51-24PI	40P6	
33	4.5V to 5.5V	AT89S51-33AC	44A	Commercial (0° C to 70° C)
		AT89S51-33JC	44J	
		AT89S51-33PC	40P6	

 = Preliminary Availability

Package Type

4A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
4J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
0P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)



Features

Low-voltage and Standard-voltage Operation

- 2.7 ($V_{CC} = 2.7V$ to $5.5V$)
- 1.8 ($V_{CC} = 1.8V$ to $5.5V$)

Internally Organized 128 x 8 (1K), 256 x 8 (2K), 512 x 8 (4K), 1024 x 8 (8K) or 2048 x 8 (16K)

Two-wire Serial Interface

Schmitt Trigger, Filtered Inputs for Noise Suppression

3-directional Data Transfer Protocol

100 kHz (1.8V) and 400 kHz (2.7V, 5V) Compatibility

Write Protect Pin for Hardware Data Protection

1-byte Page (1K, 2K), 16-byte Page (4K, 8K, 16K) Write Modes

Partial Page Writes Allowed

Self-timed Write Cycle (5 ms max)

High-reliability

- Endurance: 1 Million Write Cycles
- Data Retention: 100 Years

Automotive Grade and Lead-free/Halogen-free Devices Available

8-lead PDIP, 8-lead JEDEC SOIC, 8-lead MAP, 5-lead SOT23,

8-lead TSSOP and 8-ball dBGAA2 Packages

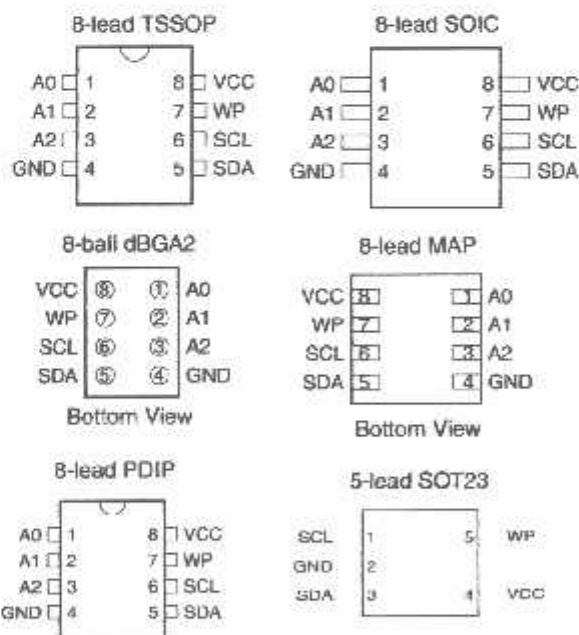
Die Sales: Wafer Form, Wafer Pack and Bumped Wafers

Description

The AT24C01A/02/04/08A/16A provides 1024/2048/4096/8192/16384 bits of serial electrically erasable and programmable read-only memory (EEPROM) organized as 8/256/512/1024/2048 words of 8 bits each. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation is essential. The AT24C01A/02/04/08A/16A is available in space-saving 8-lead PDIP, 8-lead JEDEC SOIC, 8-lead MAP, 5-lead SOT23 (AT24C01A/AT24C02/AT24C04), 8-lead TSSOP, and 8-ball dBGAA2 packages and is accessed via a Two-wire serial interface. In addition, the entire family is available in 2.7V (2.7V to 5.5V) and 1.8V (1.8V to 5V) versions.

Table 1. Pin Configuration

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
NC	No Connect
GND	Ground
VCC	Power Supply



Two-wire Serial EEPROM

1K (128 x 8)

2K (256 x 8)

4K (512 x 8)

8K (1024 x 8)

16K (2048 x 8)

AT24C01A

AT24C02

AT24C04

AT24C08A

AT24C16A

0180V-SEEPR-8/05

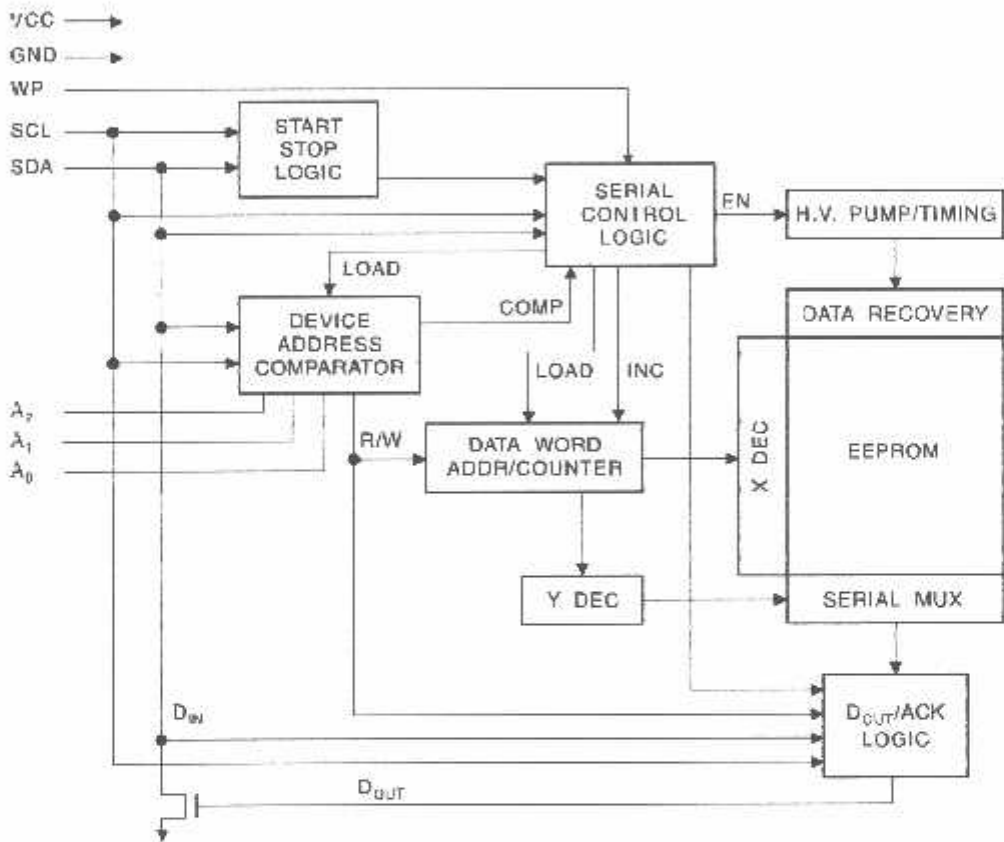


Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.25V
IO Output Current	5.0 mA

***NOTICE:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Figure 1. Block Diagram



n Description

SERIAL CLOCK (SCL): The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

SERIAL DATA (SDA): The SDA pin is bidirectional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open-collector devices.

DEVICE/PAGE ADDRESSES (A2, A1, A0): The A2, A1 and A0 pins are device address inputs that are hard wired for the AT24C01A and the AT24C02. As many as eight 1K/2K devices may be addressed on a single bus system (device addressing is discussed in detail under the Device Addressing section).

The AT24C04 uses the A2 and A1 inputs for hard wire addressing and a total of four 4K devices may be addressed on a single bus system. The A0 pin is a no connect.

The AT24C08A only uses the A2 input for hardwire addressing and a total of two 8K devices may be addressed on a single bus system. The A0 and A1 pins are no connects.

The AT24C16A does not use the device address pins, which limits the number of devices on a single bus to one. The A0, A1 and A2 pins are no connects.

WRITE PROTECT (WP): The AT24C01A/02/04/08A/16A has a Write Protect pin that provides hardware data protection. The Write Protect pin allows normal Read/Write operations when connected to ground (GND). When the Write Protect pin is connected to V_{CC}, the write protection feature is enabled and operates as shown in Table 2.

Table 2. Write Protect

WP Pin Status	Part of the Array Protected				
	24C01A	24C02	24C04	24C08A	24C16A
At V _{CC}	Full (1K) Array	Full (2K) Array	Full (4K) Array	Full (8K) Array	Full (16K) Array
At GND	Normal Read/Write Operations				

Memory Organization

AT24C01A, 1K SERIAL EEPROM: Internally organized with 16 pages of 8 bytes each, the 1K requires a 7-bit data word address for random word addressing.

AT24C02, 2K SERIAL EEPROM: Internally organized with 32 pages of 8 bytes each, the 2K requires an 8-bit data word address for random word addressing.

AT24C04, 4K SERIAL EEPROM: Internally organized with 32 pages of 16 bytes each, the 4K requires a 9-bit data word address for random word addressing.

AT24C08A, 8K SERIAL EEPROM: Internally organized with 64 pages of 16 bytes each, the 8K requires a 10-bit data word address for random word addressing.

AT24C16A, 16K SERIAL EEPROM: Internally organized with 128 pages of 16 bytes each, the 16K requires an 11-bit data word address for random word addressing.

Table 3. Pin Capacitance⁽¹⁾

Applicable over recommended operating range from $T_A = 25^{\circ}\text{C}$, $f = 1.0\text{ MHz}$, $V_{CC} = +1.8\text{V}$

Symbol	Test Condition	Max	Units	Conditions
C_{IO}	Input/Output Capacitance (SDA)	8	pF	$V_{IO} = 0\text{V}$
C_{IN}	Input Capacitance (A_0 , A_1 , A_2 , SCL)	6	pF	$V_{IN} = 0\text{V}$

Note: 1. This parameter is characterized and is not 100% tested.

Table 4. DC Characteristics

Applicable over recommended operating range from: $T_{AI} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $V_{CC} = +1.8\text{V}$ to $+5.5\text{V}$, $V_{CC} = +1.8\text{V}$ to $+5.5\text{V}$ (less otherwise noted)

Symbol	Parameter	Test Condition	Min	Typ	Max	Units
V_{CC1}	Supply Voltage		1.8		5.5	V
V_{CC2}	Supply Voltage		2.7		5.5	V
V_{CC3}	Supply Voltage		4.5		5.5	V
I_C	Supply Current $V_{CC} = 5.0\text{V}$	READ at 100 kHz		0.4	1.0	mA
I_C	Supply Current $V_{CC} = 5.0\text{V}$	WRITE at 100 kHz		2.0	3.0	mA
I_{B1}	Standby Current $V_{CC} = 1.8\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}		0.6	3.0	μA
I_{B2}	Standby Current $V_{CC} = 2.5\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}		1.4	4.0	μA
I_{B3}	Standby Current $V_{CC} = 2.7\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}		1.6	4.0	μA
I_{B4}	Standby Current $V_{CC} = 5.0\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}		8.0	18.0	μA
I_I	Input Leakage Current	$V_{IN} = V_{CC}$ or V_{SS}		0.10	3.0	μA
I_O	Output Leakage Current	$V_{OUT} = V_{CC}$ or V_{SS}		0.05	3.0	μA
V_{IL}	Input Low Level ⁽¹⁾		-0.6		$V_{CC} \times 0.3$	V
V_{IH}	Input High Level ⁽¹⁾		$V_{CC} \times 0.7$		$V_{CC} + 0.5$	V
V_{OL2}	Output Low Level $V_{CC} = 3.0\text{V}$	$I_{OL} = 2.1\text{ mA}$			0.4	V
V_{OL1}	Output Low Level $V_{CC} = 1.8\text{V}$	$I_{OL} = 0.15\text{ mA}$			0.2	V

Note: 1. V_{IL} min and V_{IH} max are reference only and are not tested.

Table 5. AC Characteristics
Applicable over recommended operating range from $T_{AI} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $V_{CC} = +1.8\text{V}$ to $+5.5\text{V}$, $V_{CC} = +2.7\text{V}$ to $+5.5\text{V}$, $C_L = 100\text{ pF}$ and 1 TTL Gate and 100 pF (unless otherwise noted)

Symbol	Parameter	1.8-volt		2.7, 5.0-volt		Units
		Min	Max	Min	Max	
f_{CL}	Clock Frequency, SCL		100		400	kHz
t_{CW}	Clock Pulse Width Low	4.7		1.2		μs
t_{CH}	Clock Pulse Width High	4.0		0.6		μs
	Noise Suppression Time ⁽¹⁾		100		50	ns
t_{A}	Clock Low to Data Out Valid	0.1	4.5	0.1	0.9	μs
t_{DF}	Time the bus must be free before a new transmission can start ⁽¹⁾	4.7		1.2		μs
$t_{D,STA}$	Start Hold Time	4.0		0.6		μs
$t_{U,STA}$	Start Setup Time	4.7		0.6		μs
$t_{D,DAT}$	Data In Hold Time	0		0		μs
$t_{U,DAT}$	Data In Setup Time	200		100		ns
t_{IR}	Inputs Rise Time ⁽¹⁾		1.0		0.3	μs
	Inputs Fall Time ⁽¹⁾		300		300	ns
$t_{SU,STO}$	Stop Setup Time	4.7		0.6		μs
t_{DH}	Data Out Hold Time	100		50		ns
t_{WR}	Write Cycle Time		5		5	ms
Endurance ⁽¹⁾	5.0V, 25°C, Byte Mode	1M		1M		Write Cycles

Note: 1. This parameter is characterized.



Device Operation

CLOCK and DATA TRANSITIONS: The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods (see Figure 4 on page 7). Data changes during SCL high periods will indicate a start or stop condition as defined below.

START CONDITION: A high-to-low transition of SDA with SCL high is a start condition which must precede any other command (see Figure 5 on page 8).

STOP CONDITION: A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the stop command will place the EEPROM in a standby power mode (see Figure 5 on page 8).

ACKNOWLEDGE: All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero to acknowledge that it has received each word. This happens during the ninth clock cycle.

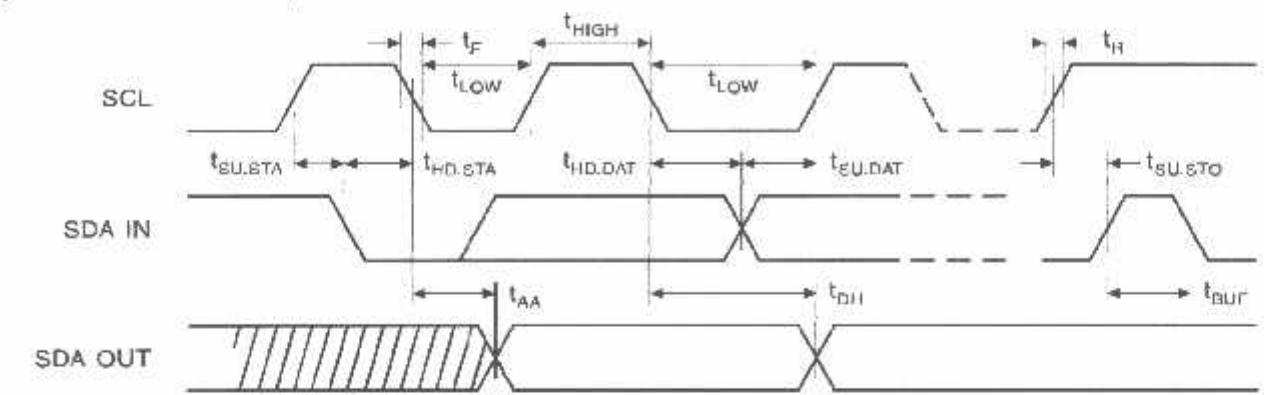
STANDBY MODE: The AT24C01A/02/04/08A/16A features a low-power standby mode which is enabled: (a) upon power-up and (b) after the receipt of the STOP bit and the completion of any internal operations.

MEMORY RESET: After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

1. Clock up to 9 cycles.
2. Look for SDA high in each cycle while SCL is high.
3. Create a start condition.

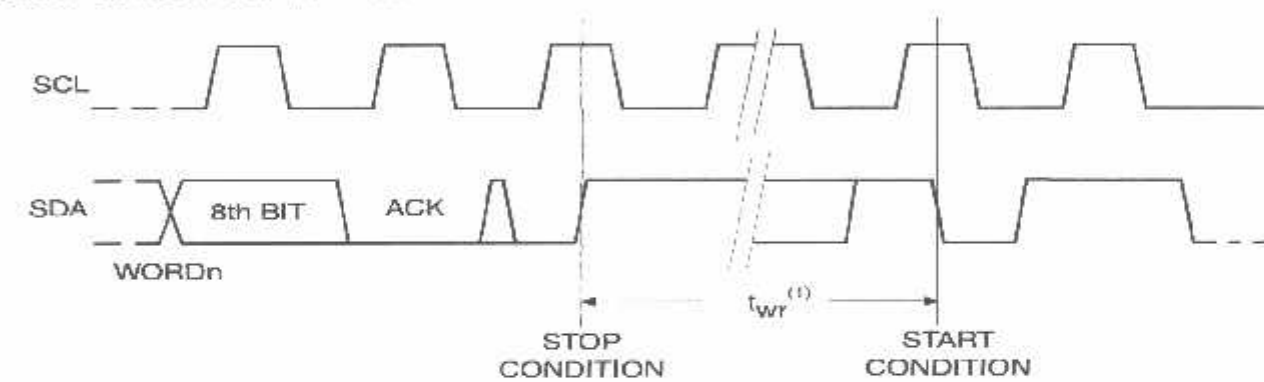
Timing

Figure 2. SCL: Serial Clock, SDA: Serial Data I/O



Write Cycle Timing

Figure 3. SCL: Serial Clock, SDA: Serial Data I/O



Note: 1. The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

Figure 4. Data Validity

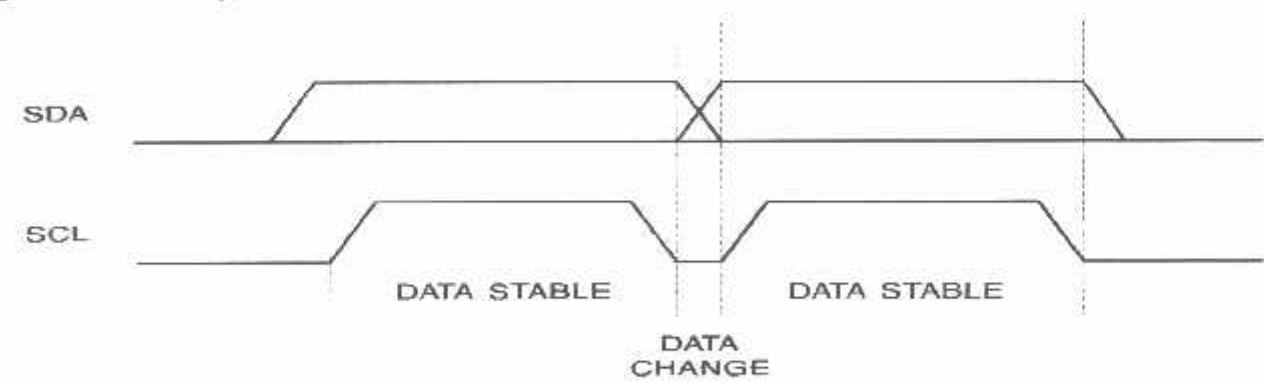


Figure 5. Start and Stop Definition

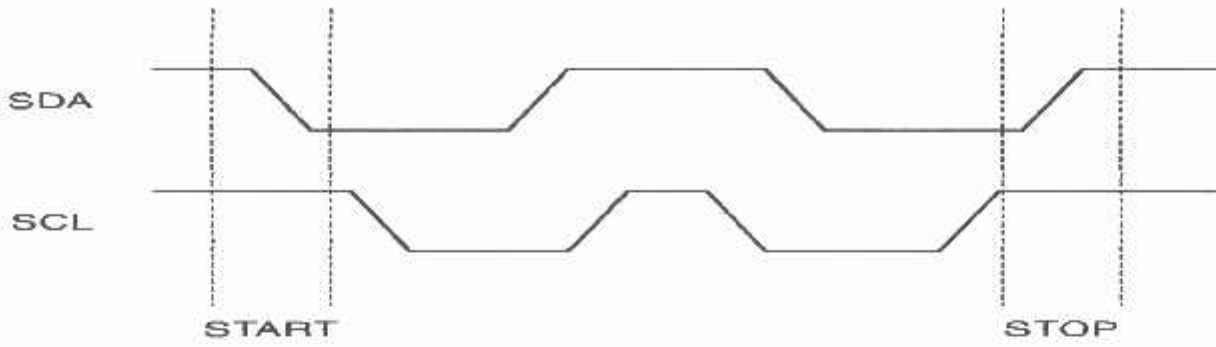
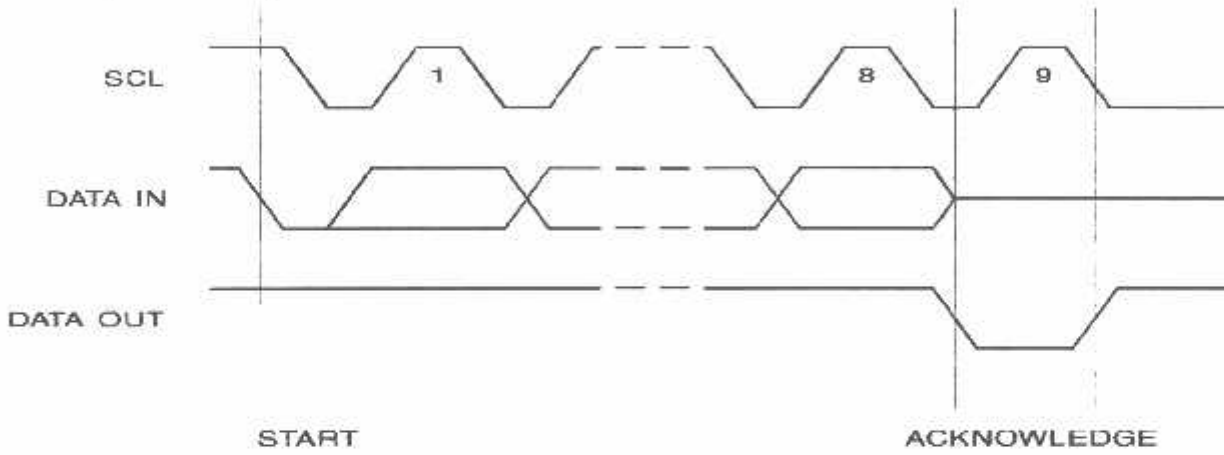


Figure 6. Output Acknowledge



Device Addressing

The 1K, 2K, 4K, 8K and 16K EEPROM devices all require an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 7).

The device address word consists of a mandatory one, zero sequence for the first four most significant bits as shown. This is common to all the EEPROM devices.

The next 3 bits are the A2, A1 and A0 device address bits for the 1K/2K EEPROM. These 3 bits must compare to their corresponding hard-wired input pins.

The 4K EEPROM only uses the A2 and A1 device address bits with the third bit being a memory page address bit. The two device address bits must compare to their corresponding hard-wired input pins. The A0 pin is no connect.

The 8K EEPROM only uses the A2 device address bit with the next 2 bits being for memory page addressing. The A2 bit must compare to its corresponding hard-wired input pin. The A1 and A0 pins are no connect.

The 16K does not use any device address bits but instead the 3 bits are used for memory page addressing. These page addressing bits on the 4K, 8K and 16K devices should be considered the most significant bits of the data word address which follows. The A0, A1 and A2 pins are no connect.

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

Upon a compare of the device address, the EEPROM will output a zero. If a compare is not made, the chip will return to a standby state.

Write Operations

BYTE WRITE: A write operation requires an 8-bit data word address following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero and the addressing device, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally timed write cycle, t_{WR} , to the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (see Figure 8 on page 11).

PAGE WRITE: The 1K/2K EEPROM is capable of an 8-byte page write, and the 4K, 8K and 16K devices are capable of 16-byte page writes.

A page write is initiated the same as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to seven (1K/2K) or fifteen (4K, 8K, 16K) more data words. The EEPROM will respond with a zero after each data word received. The microcontroller must terminate the page write sequence with a stop condition (see Figure 9 on page 11).

The data word address lower three (1K/2K) or four (4K, 8K, 16K) bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than eight (1K/2K) or sixteen (4K, 8K, 16K) data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten.





ACKNOWLEDGE POLLING: Once the internally timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero allowing the read or write sequence to continue.

Read operations

Read operations are initiated the same way as write operations with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

CURRENT ADDRESS READ: The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "roll over" during read is from the last byte of the last memory page to the first byte of the first page. The address "roll over" during write is from the last byte of the current page to the first byte of the same page.

Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero but does generate a following stop condition (see Figure 10 on page 12).

RANDOM READ: A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller does not respond with a zero but does generate a following stop condition (see Figure 11 on page 12).

SEQUENTIAL READ: Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledge. As long as the EEPROM receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will "roll over" and the sequential read will continue. The sequential read operation is terminated when the microcontroller does not respond with a zero but does generate a following stop condition (see Figure 12 on page 12).

Figure 7. Device Address



Figure 8. Byte Write

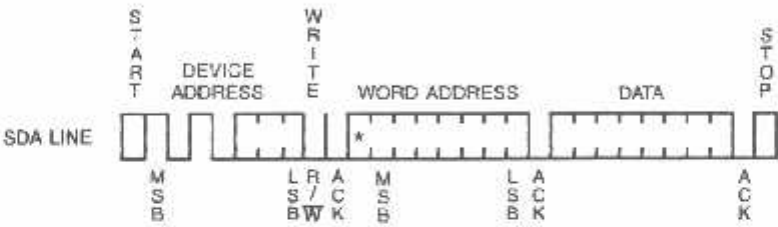
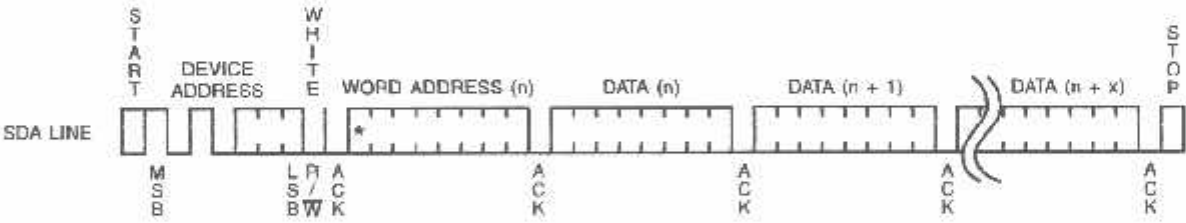


Figure 9. Page Write



(* = DONT CARE bit for 1K)

Figure 10. Current Address Read

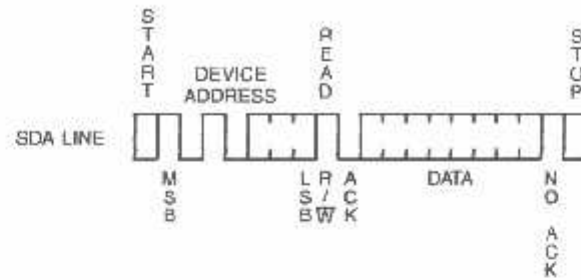
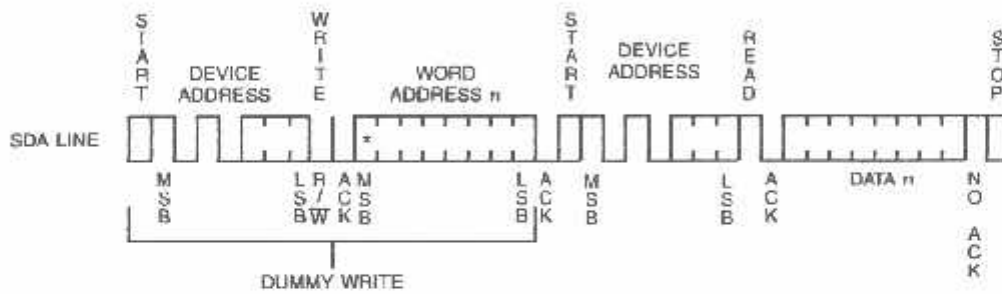
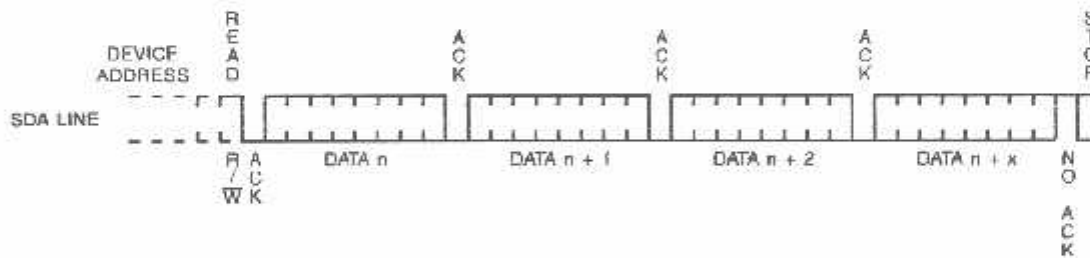


Figure 11. Random Read



(* = DON'T CARE bit for 1K)

Figure 12. Sequential Read



AT24C01A Ordering Information⁽¹⁾

Ordering Code	Package	Operation Range
AT24C01A-10PI-2.7 AT24C01A-10SI-2.7 AT24C01A-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C01A-10PI-1.8 AT24C01A-10SI-1.8 AT24C01A-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C01A-10PU-2.7 ⁽²⁾ AT24C01A-10PU-1.8 ⁽²⁾ AT24C01A-10SU-2.7 ⁽²⁾ AT24C01A-10SU-1.8 ⁽²⁾ AT24C01A-10TU-2.7 ⁽²⁾ AT24C01A-10TU-1.8 ⁽²⁾ AT24C01A-10TSU-1.8 ⁽²⁾ AT24C01AU3-10UU-1.8 ⁽²⁾ AT24C01AY1-10YU-1.8 ⁽²⁾	8P3 8P3 8S1 8S1 8A2 8A2 5TS1 8U31 8Y1	Lead-free/Halogen-free/ Industrial Temperature (-40°C to 85°C)
AT24C01A-W2.7-11 ⁽³⁾ AT24C01A-W1.8-11 ⁽³⁾	Die Sale Die Sale	Industrial Temperature (-40°C to 85°C)

- Notes:
- 1. For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics table.
 - 2. "U" designates Green Package + RoHS compliant.
 - 3. Available in wafer pack and wafer form; order as SL719 for wafer form. Bumped die available upon request. Please contact Serial EEPROM Marketing.

Package Type	
8P3	8-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)
8Y1	8-lead, 4.90 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
5TS1	5-lead, 2.90 mm x 1.60 mm Body, Plastic Thin Shrink Small Outline Package (SOT23)
8U3-1	8-ball, die Ball Grid Array Package (dBGAA2)
Options	
-2.7	Low-voltage (2.7V to 5.5V)
-1.8	Low-voltage (1.8V to 5.5V)



AT24C02 Ordering Information⁽¹⁾

Ordering Code	Package	Operation Range
AT24C02-10PI-2.7 AT24C02N-10SI-2.7 AT24C02-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C02-10PI-1.8 AT24C02N-10SI-1.8 AT24C02-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C02-10PU-2.7 ⁽²⁾ AT24C02-10PU-1.8 ⁽²⁾ AT24C02N-10SU-2.7 ⁽²⁾ AT24C02N-10SU-1.8 ⁽²⁾ AT24C02-10TU-2.7 ⁽²⁾ AT24C02-10TU-1.8 ⁽²⁾ AT24C02Y1-10YU-1.8 ⁽²⁾ AT24C02-10TSU-1.8 ⁽²⁾ AT24C02U3-10UU-1.8 ⁽²⁾	8P3 8P3 8S1 8S1 8A2 8A2 8Y1 5TS1 8U3-1	Lead-free/Halogen-free/ Industrial Temperature (-40°C to 85°C)
AT24C02-W2.7-11 ⁽³⁾ AT24C02-W1.8-11 ⁽³⁾	Die Sale Die Sale	Industrial Temperature (-40°C to 85°C)

- Notes: 1. For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics table.
2. "U" designates Green Package + RoHS compliant.
3. Available in wafer pack and wafer form; order as SL719 for wafer form. Bumped die available upon request. Please contact Serial EEPROM Marketing.

Package Type	
8P3	8-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)
8Y1	8-lead, 4.90 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
5TS1	5-lead, 2.90 mm x 1.60 mm Body, Plastic Thin Shrink Small Outline Package (SOT23)
8U3-1	8-ball, die Ball Grid Array Package (dBGAA2)
Options	
-2.7	Low-voltage (2.7V to 5.5V)
-1.8	Low-voltage (1.8V to 5.5V)

AT24C04 Ordering Information⁽¹⁾

Ordering Code	Package	Operation Range
AT24C04-10PI-2.7 AT24C04N-10SI-2.7 AT24C04-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C04-10PI-1.8 AT24C04N-10SI-1.8 AT24C04-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C04-10PU-2.7 ⁽²⁾ AT24C04-10PU-1.8 ⁽²⁾ AT24C04N-10SU-2.7 ⁽²⁾ AT24C04N-10SU-1.8 ⁽²⁾ AT24C04-10TU-2.7 ⁽²⁾ AT24C04-10TU-1.8 ⁽²⁾ AT24C04Y1-10YU-1.8 ⁽²⁾ AT24C04-10TSU-1.8 ⁽²⁾ AT24C04U3-10UU-1.8 ⁽²⁾	8P3 8P3 8S1 8S1 8A2 8A2 8Y1 5TS1 8U3-1	Lead-free/Halogen-free/ Industrial Temperature (-40°C to 85°C)
AT24C04-W2.7-11 ⁽³⁾ AT24C04-W1.8-11 ⁽³⁾	Die Sale Die Sale	Industrial Temperature (-40°C to 85°C)

- Notes: 1. For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics table.
2. "U" designates Green Package + RoHS compliant.
3. Available in wafer pack and wafer form; order as SL719 for wafer form. Bumped die available upon request. Please contact Serial EEPROM Marketing.

Package Type	
8P3	8-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)
8Y1	8-lead, 4.90 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
5TS1	5-lead, 2.90 mm x 1.60 mm Body, Plastic Thin Shrink Small Outline Package (SOT23)
8U3-1	8-ball, die Ball Grid Array Package (dBGAA2)
Options	
2.7	Low-voltage (2.7V to 5.5V)
1.8	Low-voltage (1.8V to 5.5V)





AT24C08A Ordering Information⁽¹⁾

Ordering Code	Package	Operation Range
AT24C08A-10PI-2.7 AT24C08AN-10SI-2.7 AT24C08A-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C08A-10PI-1.8 AT24C08AN-10SI-1.8 AT24C08A-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
AT24C08A-10PU-2.7 ⁽²⁾ AT24C08A-10PU-1.8 ⁽²⁾ AT24C08AN-10SU-2.7 ⁽²⁾ AT24C08AN-10SU-1.8 ⁽²⁾ AT24C08A-10TU-2.7 ⁽²⁾ AT24C08A-10TU-1.8 ⁽²⁾ AT24C08AY1-10YU-1.8 ⁽²⁾ AT24C08AY5-10YU-1.8 ⁽²⁾ AT24C08AU2-10UU-1.8 ⁽²⁾	8P3 8P3 8S1 8S1 8A2 8A2 8Y1 8Y5 8U2-1	Lead-free/Halogen-free/ Industrial Temperature (-40°C to 85°C)
AT24C08A-W2 7-11 ⁽³⁾ AT24C08A-W1 8-11 ⁽³⁾	Die Sale Die Sale	Industrial Temperature (-40°C to 85°C)

- tes: 1. For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics table.
2. "U" designates Green Package + RoHS compliant.
3. Available in waffle pack and wafer form; order as SL719 for wafer form. Bumped die available upon request. Please contact Serial EEPROM Marketing.

Package Type	
8P3	8-pin, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)
8Y1	8-lead, 4.90 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
8Y5	8-lead, 2.00 mm x 3.00 mm Body, Dual Footprint, Non-Leaded, Miniature Array Package (MAP)
8U2-1	8-ball, die Ball Grid Array Package (dBGAA2)
Options	
-2.7	Low Voltage (2.7V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)

AT24C16A Ordering Information⁽¹⁾

Ordering Code	Package	Operation Range
AT24C16A-10PI-2.7	8P3	Industrial Temperature (-40°C to 85°C)
AT24C16AN-10SI-2.7	8S1	
AT24C16A-10TI-2.7	8A2	
AT24C16A-10PI-1.8	8P3	Industrial Temperature (-40°C to 85°C)
AT24C16AN-10SI-1.8	8S1	
AT24C16A-10TI-1.8	8A2	
AT24C16A-10PU-2.7 ⁽²⁾	8P3	Lead-free/Halogen-free/ Industrial Temperature (-40°C to 85°C)
AT24C16A-10PU-1.8 ⁽²⁾	8P3	
AT24C16AN-10SU-2.7 ⁽²⁾	8S1	
AT24C16AN-10SU-1.8 ⁽²⁾	8S1	
AT24C16A-10TU-2.7 ⁽²⁾	8A2	
AT24C16A-10TU-1.8 ⁽²⁾	8A2	
AT24C16AY1-10YU-1.8 ⁽²⁾	8Y1	
AT24C16AY5-10YU-1.8 ⁽²⁾	8Y5	
AT24C16AU2-10UU-1.8 ⁽²⁾	8U2-1	
AT24C16A-W2 7-11 ⁽³⁾	Die Sale	Industrial Temperature (-40°C to 85°C)
AT24C16A-W1 8-11 ⁽³⁾	Die Sale	

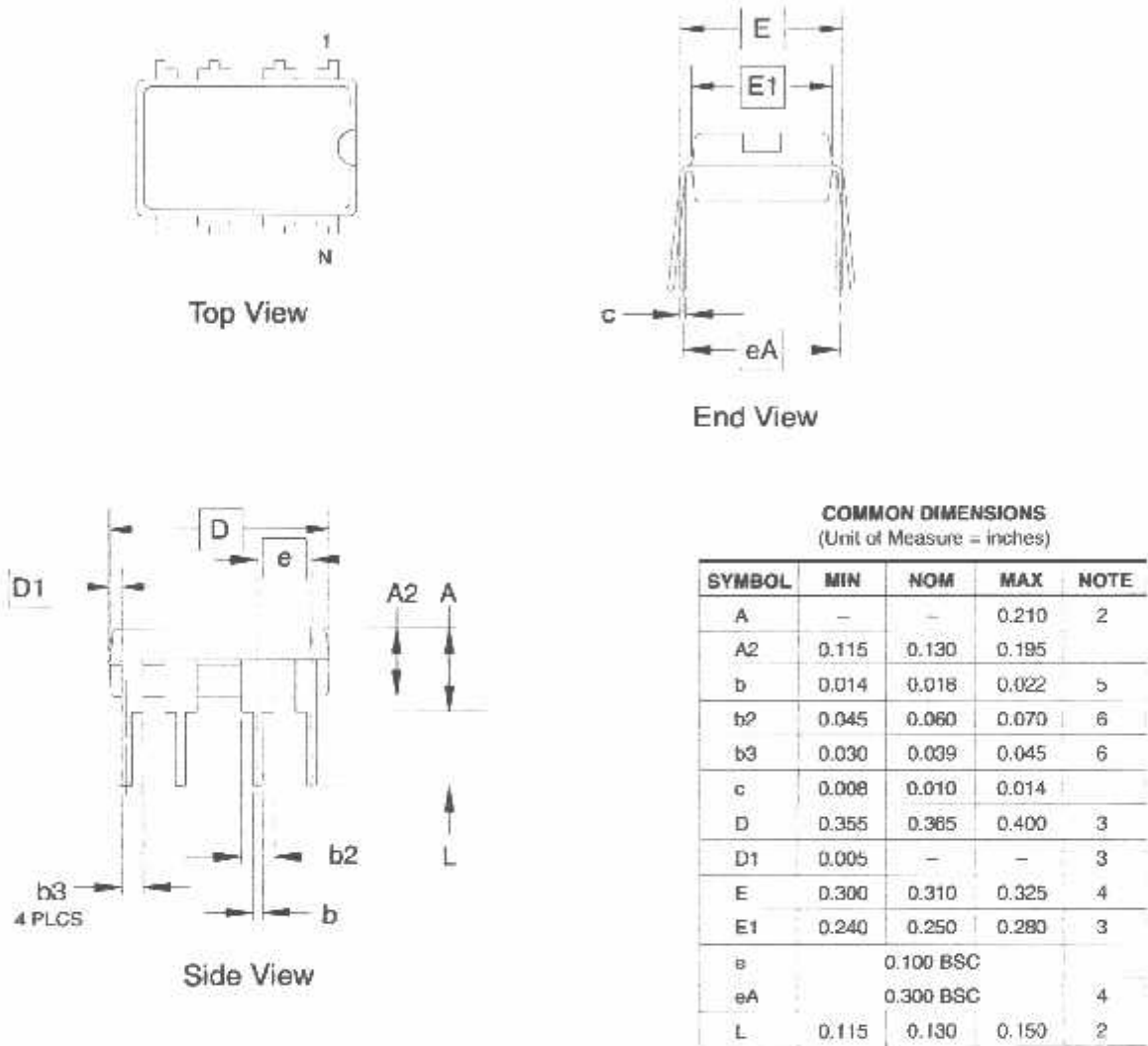
- Notes: 1. For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics table.
2. "U" designates Green Package + RoHS compliant
3. Available in wafer pack and wafer form; order as SL719 for wafer form. Bumped die available upon request. Please contact Serial EEPROM Marketing.

Package Type	
8P3	8-pin, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8S1	8-lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8A2	8-lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
8Y1	8-lead, 4.90 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
8Y5	8-lead, 2.00 mm x 3.00 mm Body, Dual Footprint, Non-leaded, Miniature Array Package (MAP)
8U2-1	8-ball, die Ball Grid Array Package (dBGAA2)
Options	
-2.7	Low Voltage (2.7V to 5.5V)
-1.8	Low Voltage (1.8V to 5.5V)



Packaging Information

8P3 – PDIP

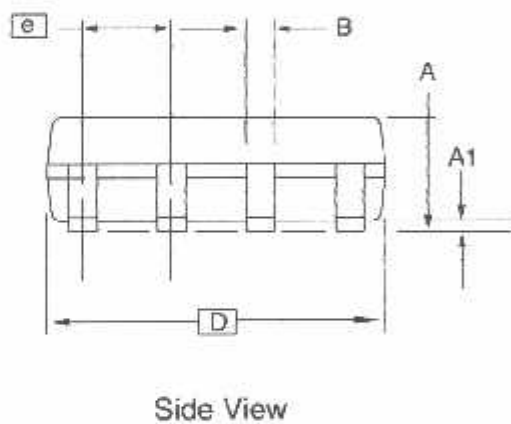
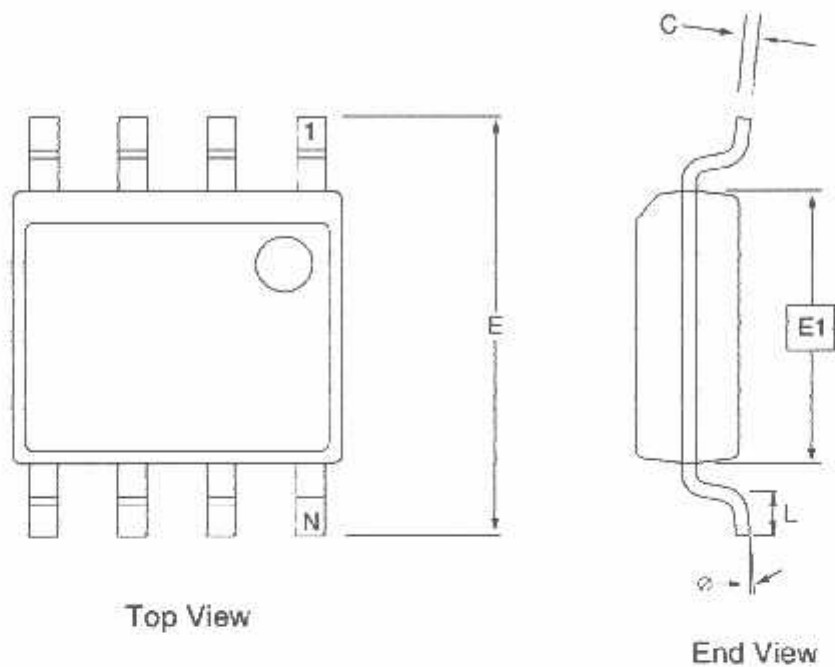


- Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-001, Variation BA, for additional information.
 2. Dimensions A and L are measured with the package seated in JEDEC seating plane Gauge GS-3.
 3. D, D1 and E1 dimensions do not include mold Flash or protrusions. Mold Flash or protrusions shall not exceed 0.010 inch.
 4. E and eA measured with the leads constrained to be perpendicular to datum.
 5. Pointed or rounded lead tips are preferred to ease insertion.
 6. b2 and b3 maximum dimensions do not include Dambar protrusions. Dambar protrusions shall not exceed 0.010 (0.25 mm).

01/09/02

2325 Orchard Parkway San Jose, CA 95131	TITLE 8P3, 8-lead, 0.300" Wide Body, Plastic Dual In-line Package (PDIP)	DRAWING NO. 8P3	REV. B
--	---	---------------------------	------------------

1 – JEDEC SOIC




COMMON DIMENSIONS
(Unit of Measure = mm)

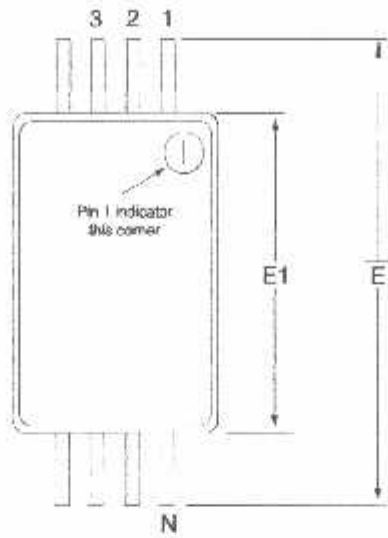
SYMBOL	MIN	NOM	MAX	NOTE
A	1.35	—	1.75	
A1	0.10	—	0.25	
b	0.31	—	0.51	
C	0.17	—	0.25	
D	4.80	—	5.00	
E1	3.81	—	3.99	
E	5.79	—	6.20	
e	1.27 BSC			
L	0.40	—	1.27	
Ø	0°	—	6°	

Note: These drawings are for general information only. Refer to JEDEC Drawing MS-012, Variation AA for proper dimensions, tolerances, datums, etc.

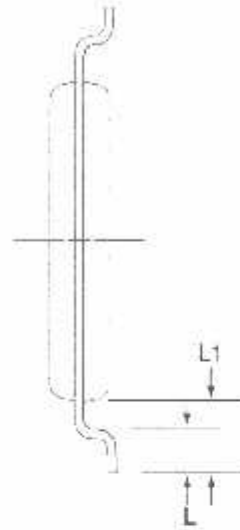
10/7/03

 1150 E. Cheyenne Mtn. Blvd. Colorado Springs, CO 80906	TITLE 8S1, 8-lead (0.150" Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)	DRAWING NO. 8S1	REV. B

2 – TSSOP



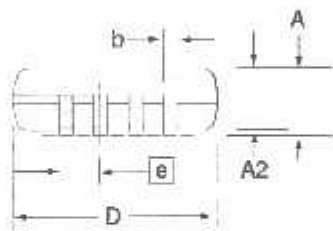
Top View



End View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
D	2.90	3.00	3.10	2, 5
E	6.40 BSC			
E1	4.30	4.40	4.50	3, 5
A	—	—	1.20	
A2	0.80	1.00	1.05	
b	0.19	—	0.30	4
e	0.65 BSC			
L	0.45	0.60	0.75	
L1	1.00 REF			



Side View

- Notes:
1. This drawing is for general information only. Refer to JEDEC Drawing MO-153, Variation AA, for proper dimensions, tolerances, datums, etc.
 2. Dimension D does not include mold flash, protrusions or gate burrs. Mold flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006 in) per side.
 3. Dimension E1 does not include inter-lead flash or protrusions. Inter-lead flash and protrusions shall not exceed 0.25 mm (0.010 in) per side.
 4. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08 mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius of the foot. Minimum space between protrusion and adjacent lead is 0.07 mm.
 5. Dimension D and E1 to be determined at Datum Plane H.

5/30/02



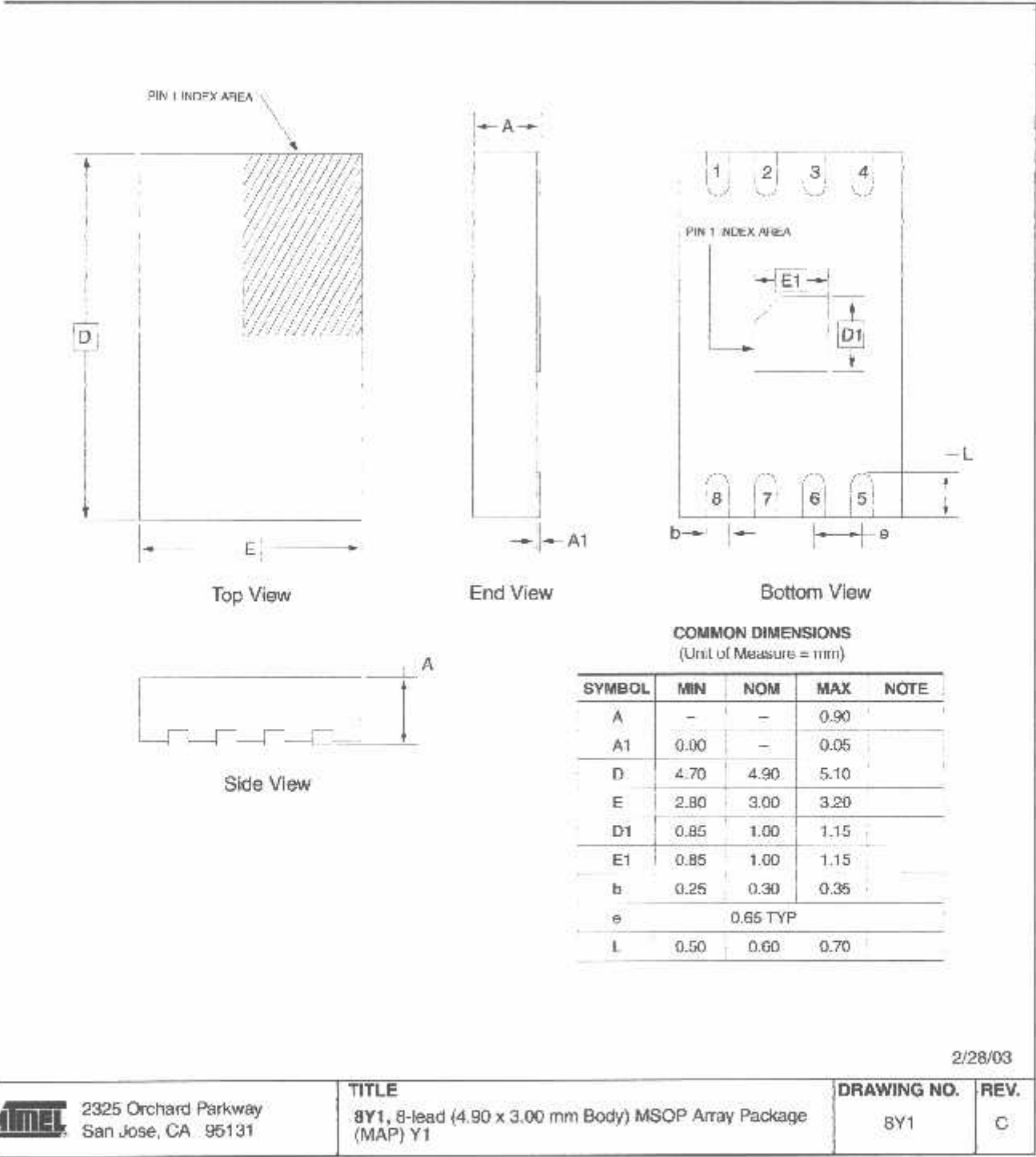
2325 Orchard Parkway
San Jose, CA 95131

TITLE
8A2, 8-lead, 4.4 mm Body, Plastic
Thin Shrink Small Outline Package (TSSOP)

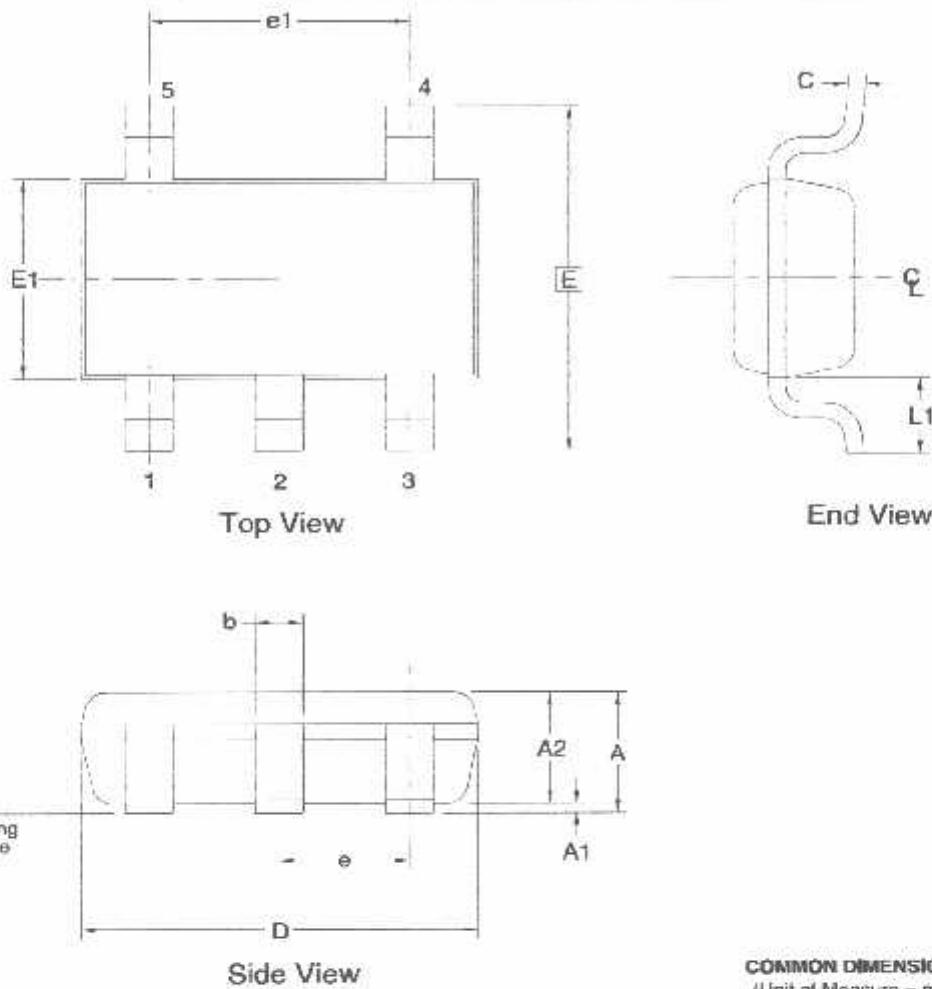
DRAWING NO.
8A2

REV.
B

Y1 – MAP



'S1 – SOT23



- NOTES:
1. This drawing is for general information only. Refer to JEDEC Drawing MC-193, Variation AB, for additional information.
 2. Dimension D does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed 0.15 mm per end. Dimension $E1$ does not include interlead flash or protrusion. Interlead flash or protrusion shall not exceed 0.15 mm per side.
 3. The package top may be smaller than the package bottom. Dimensions D and $E1$ are determined at the outermost extremes of the plastic body exclusive of mold flash, tie bar burrs, gate burrs, and interlead flash, but including any mismatch between the top and bottom of the plastic body.
 4. These dimensions apply to the flat section of the lead between 0.08 mm and 0.15 mm from the lead tip.
 5. Dimension "b" does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08 mm total in excess of the "b" dimension at maximum material condition. The Dambar cannot be located on the lower radius of the foot. Minimum space between protrusion and an adjacent lead shall not be less than 0.07 mm.

COMMON DIMENSIONS (Unit of Measure – mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.10	
A1	0.00	–	0.10	
A2	0.70	0.90	1.00	
c	0.08	–	0.20	4
D	2.90 BSC			2, 3
E	2.80 BSC			2, 3
E1	1.60 BSC			2, 3
L1	0.60 REF			
e	0.95 BSC			
e1	1.90 BSC			
b	0.30	–	0.50	4, 5

6/25/03



1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906

TITLE

5TS1, 5-lead, 1.60 mm Body, Plastic Thin Shrink
Small Outline Package (SHRINK SOT)

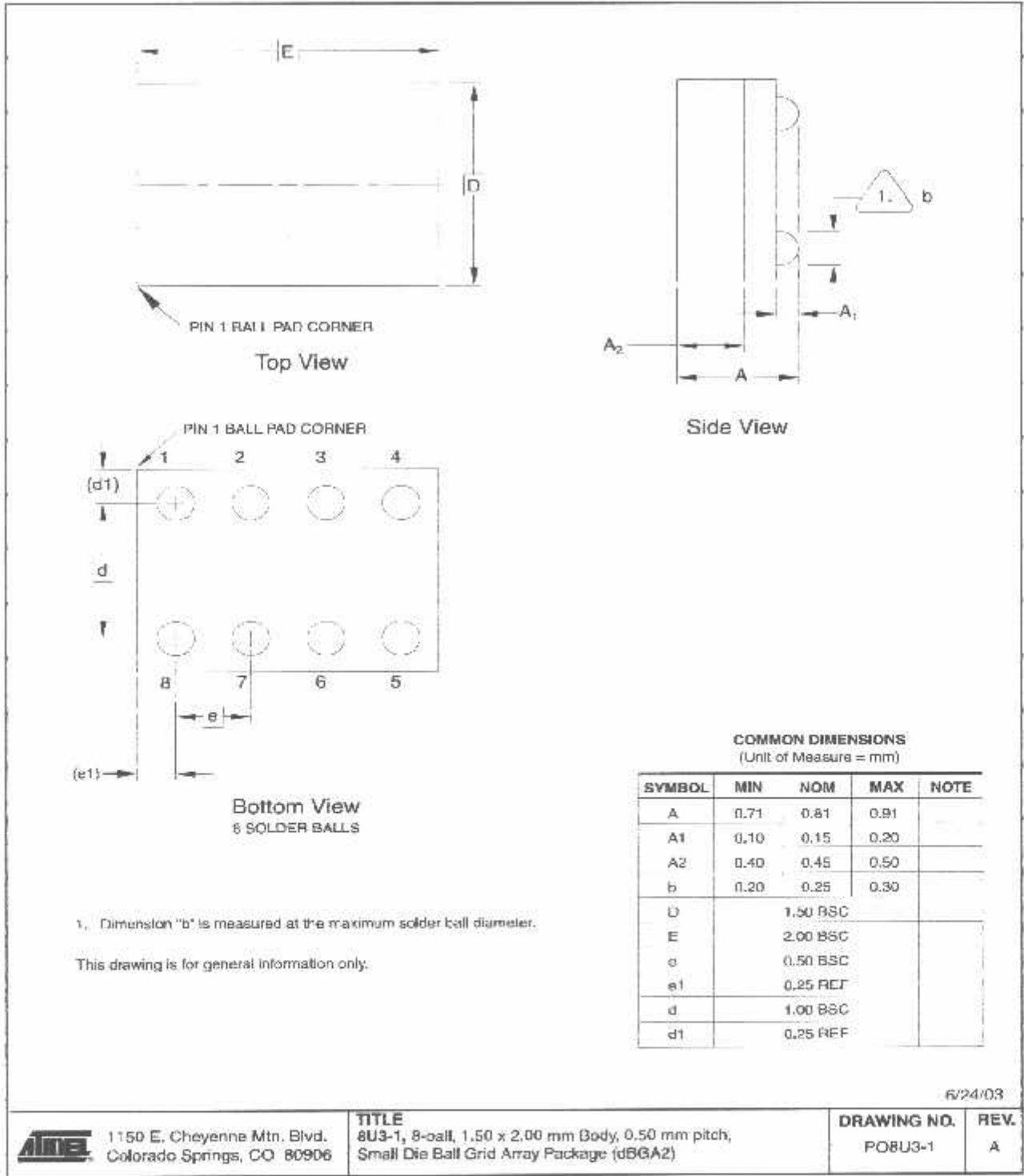
DRAWING NO.

POSTS1

REV.

A

J3-1 – dBGA2





Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 437-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenalux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantretrie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

**Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom**
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use in components in applications intended to support or sustain life.

Atmel Corporation 2005. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are® and others, are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Printed on recycled paper.

0180V SEEPR-B05

LIGHT METER

Model : LX-1102

ISO-9001, CE, IEC1010

FEATURES

- * 5 ranges :
40.00/400.0/4,000/40,000/400,000 Lux.
- * Resolution :
0.01 Lux to 100 Lux.
0.01 Ft-cd to 10 Ft-cd.
- * Sensor used the exclusive photo diode
& color correction filter, spectrum
meet C.I.E. photopic.
- * Sensor cosine correction factor
meet standard.
- * Separate Light Sensor allows user
to measure the light at an optimum
position.
- * Lux & Foot-candle unit selection.
- * Large LCD display with bar graph.
- * Water resistance front panel
- * Zero adjusting button.
- * Data hold and Peak hold.
- * Memory function with Recall. Auto
- * Auto power off or manual power off.
- * RS-232 computer data output.



LT Lutron

LUTRON ELECTRONIC

The Art of Measurement

LIGHT METER

Model : LX-1102

FEATURES

- * 5 ranges :
40.00/400.0/4,000/40,000/400,000 Lux,
wide measuring range.
- * High resolution :
0.01 Lux to 100 Lux, 0.01 Ft-cd to 10 Ft-cd
- * Sensor used the exclusive photo diode &
color correction filter, spectrum meet C.I.E.
photopic.
- * Sensor cosine correction factor meet
standard.
- * Separate Light Sensor allows user to
measure the light at an optimum position.
- * Lux & Foot-candle unit selection.
- * Large LCD display with bargraph and
indicator.
- * Water resistance front panel
- * Use the rubber key for the function select.
- * LSI circuit provides high reliability and
durability.
- * Zero adjusting button.
- * Data hold.
- * Memory function to display the max. & min.
display value with Recall.
- * Auto power auto off or manual power off.
- * Compact size and excellent operation.
- * Built-in low battery indicator.
- * RS-232 computer serial data output.

GENERAL SPECIFICATIONS

Display	Large LCD display. LCD size 52 x 38 mm, 4 digits. with bargraph indicator.	Operating Temperature	0 to 50 °C (32 to 122 °F).
Measurement & ranges	5 ranges : 40.00/400.0 Lux, 4,000/40,000/400,000 Lux,	Operating Humidity	Max. 80% RH.
Unit	Lux, Foot-candle (Ft-cd).	Power Supply	DC 9V battery. 006P, MN1604 (PP3) or equivalent.
Sensor	The exclusive photo diode & color correction filter, spectrum meet C.I.E. Cosine correction factor meet standard.	Power Consumption	Approx. DC 8 mA.
Zero Adjustment	External adjustment by pushing button. (40.00 Lux range only)	Weight	220 g/0.48 LB.
Peak Hold	To hold the peak display.	Dimension	<i>Main instrument :</i> 200 x 68 x 30 mm (7.9 x 2.7 x 1.2 inch).
Data Hold	To freeze the display value.		<i>Sensor probe :</i> 82 x 55 x 7 mm (3.2 x 2.2 x 0.3 inch).
Memory	Save the max. & min. value with Recall.	Accessories Included	Instruction Manual.....1 PC. Light Sensor with protection cover.....1 PC.
Over and Under range Indication	Over range Indicator : " - - - - " Under range Indicator : " - - - - "	Optional Accessories	RS232 cable, UPCB-02 Application Software, SW-U801-WIN Hard carryingt case, CA--06 Soft carryingt case, CA--05A
Power Off	Auto or manual power off.		
Data Output	RS-232 serial data output.		

ELECTRICAL SPECIFICATIONS (23 ± 5 °C)

Measurement	Range	In-range Display	Resolution	Accuracy
Lux	40.00 Lux	0 - 40.00 Lux	0.01 Lux	± (3% rdg + 0.5 % F.S.)
	400.0 Lux	36.0 - 400.0 Lux	0.1 Lux	
	4,000 Lux	360 - 4,000 Lux	1 Lux	
	40,000 Lux	3,600 - 40,000 Lux	10 Lux	
	400,000 Lux	10,000 - 400,000 Lux	100 Lux	
				< 100,000 Lux : ± (3% rdg + 0.5 % F.S.) ≥ 100,000 Lux : @ for reference only
Foot-candle (Fc)	4.000 Fc	0 - 3.720 Fc	0.001 Fc	± (3% rdg + 0.5 % F.S.)
	40.00 Fc	3.35 - 37.20 Fc	0.01 Fc	
	400.0 Fc	33.5 - 372.0 Fc	0.1 Fc	
	4,000 Fc	335 - 3,720 Fc	1 Fc	
	40,000 Fc	930 - 37,200 Fc	10 Fc	
				< 9,300 Fc : ± (3% rdg + 0.5 % F.S.) ≥ 9,300 Fc @ for reference only

Note : Accuracy tested by a standard parallel light tungsten lamp of 2856 K degree temperature.

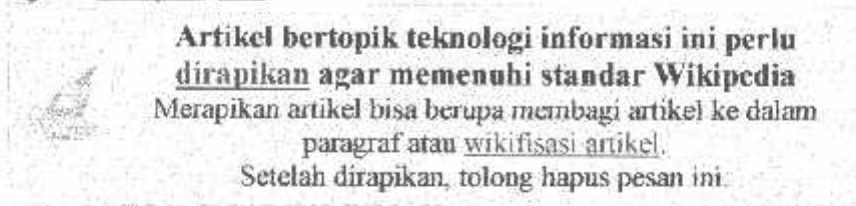
* Appearance and specifications listed in this brochure are subject to change without notice.

CAT-0302-LX1102

Port TCP dan UDP

Dari Wikipedia Indonesia, ensiklopedia bebas berbahasa Indonesia.

Langsung ke: [navigasi](#), [cari](#)



Dalam protokol jaringan **TCP/IP**, sebuah *port* adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. Port dapat mengidentifikasi aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, port juga mengidentifikasi sebuah proses tertentu di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam server. Port dapat dikenali dengan angka 16-bit (dua byte) yang disebut dengan **Port Number** dan diklasifikasikan dengan jenis protokol transport apa yang digunakan, ke dalam **Port TCP** dan **Port UDP**. Karena memiliki angka 16-bit, maka total maksimum jumlah port untuk setiap protokol transport yang digunakan adalah 65536 buah.

Dilihat dari penomorannya, port UDP dan TCP dibagi menjadi tiga jenis, yakni sebagai berikut:

- **Well-known Port**: yang pada awalnya berkisar antara 0 hingga 255 tapi kemudian diperlebar untuk mendukung antara 0 hingga 1023. Port number yang termasuk ke dalam well-known port, selalu merepresentasikan layanan jaringan yang sama, dan ditetapkan oleh Internet Assigned Number Authority (IANA). Beberapa di antara port-port yang berada di dalam range Well-known port masih belum ditetapkan dan diresevasikan untuk digunakan oleh layanan yang bakal ada di masa depan. *Well-known port* didefinisikan dalam RFC 1060.
- **Registered Port**: Port-port yang digunakan oleh vendor-vendor komputer atau jaringan yang berbeda untuk mendukung aplikasi dan sistem operasi yang mereka buat. Registered port juga diketahui dan didaftarkan oleh IANA tapi tidak dialokasikan secara permanen, sehingga vendor lainnya dapat menggunakan port number yang sama. Range registered port berkisar dari 1024 hingga 49151 dan beberapa port di antaranya adalah *Dynamically Assigned Port*.
- **Dynamically Assigned Port**: merupakan port-port yang ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan. Dynamically Assigned Port berkisar dari 1024 hingga 65536 dan dapat digunakan atau dilepaskan sesuai kebutuhan.

[sunting] Well-known Port

Tabel berikut ini berisi Well-known Port.

Port	Jenis Port	Keyword	Digunakan oleh
0	TCP, UDP	T/A	Dicadangkan, tidak digunakan.
1	TCP, UDP	TCPmux	TCP Port Service Multiplexer
2	TCP, UDP	compressnet	Management Utility
3	TCP, UDP	compressnet	Compression Process
4	TCP, UDP	T/A	Belum ditetapkan
5	TCP, UDP	rje	Remote Job Entry
6	TCP, UDP	T/A	Belum ditetapkan
7	TCP, UDP	echo	Echo
8	TCP, UDP	T/A	Belum ditetapkan
9	TCP, UDP	discard	Discard; alias=sink null
10	TCP, UDP	T/A	Belum ditetapkan
11	TCP, UDP	systat	Active Users; alias = users
12	TCP, UDP	T/A	Belum ditetapkan
13	TCP, UDP	daytime	Daytime
14	TCP, UDP	T/A	Belum ditetapkan
15	TCP, UDP	T/A	Belum ditetapkan (sebelumnya: netstat)
16	TCP, UDP	T/A	Belum ditetapkan
17	TCP, UDP	qotd	Quote of the Day; alias = quote
18	TCP, UDP	msp	Message Send Protocol
19	TCP, UDP	chargen	Character Generator; alias = ttytst source
20	TCP, UDP	ftp-data	<u>File Transfer Protocol</u> (default data)
21	TCP, UDP	ftp	<u>File Transfer Protocol</u> (control), connection dialog
22	TCP, UDP	T/A	Belum ditetapkan
23	TCP, UDP	telnet	<u>Telnet</u>
24	TCP, UDP		Any private mail system
25	TCP, UDP	smtp	<u>Simple Mail Transfer Protocol</u> ; alias = mail
26	TCP, UDP	T/A	Belum ditetapkan
27	TCP, UDP	nsw-fe	NSW User System FE
28	TCP, UDP	T/A	Belum ditetapkan
29	TCP, UDP	msg-icp	MSG ICP
30	TCP, UDP	T/A	Belum ditetapkan
31	TCP, UDP	msg-auth	MSG Authentication
32	TCP, UDP		Belum ditetapkan
33	TCP, UDP	dsp	Display Support Protocol
34	TCP, UDP	T/A	Belum ditetapkan
35	TCP, UDP		Any private printer server

36	TCP, UDP	T/A	Belum ditetapkan
37	TCP, UDP	time	Time; alias = timeserver
38	TCP, UDP	T/A	Belum ditetapkan
39	TCP, UDP	rlp	Resource Location Protocol; alias = resource
40	TCP, UDP	T/A	Belum ditetapkan
41	TCP, UDP	graphics	Graphics
42	TCP, UDP	nameserver	Host Name Server; alias = nameserver
43	TCP, UDP	nickname	Who Is; alias – nickname
44	TCP, UDP	mpm-flags	MPM FLAGS Protocol
45	TCP, UDP	mpm	Message Processing Module
46	TCP, UDP	mpm-snd	MPM (default send)
47	TCP, UDP	ni-ftp	NI FTP
48	TCP, UDP	T/A	Belum ditetapkan
49	TCP, UDP	login	Login Host Protocol
50	TCP, UDP	re-mail-ck	Remote Mail Checking Protocol
51	TCP, UDP	la-maint	IMP Logical Address Maintenance
52	TCP, UDP	xns-time	XNS Time Protocol
53	TCP, UDP	domain	<u>Domain Name System Server</u>
54	TCP, UDP	xns-ch	XNS Clearinghouse
55	TCP, UDP	isi-gl	ISI Graphics Language
56	TCP, UDP	xns-auth	XNS Authentication
57	TCP, UDP		Any private terminal access
58	TCP, UDP	xns-mail	XNS Mail
59	TCP, UDP		Any private file service
60	TCP, UDP	T/A	Belum ditetapkan
61	TCP, UDP	ni-mail	NI MAIL
62	TCP, UDP	acas	ACA Services
63	TCP, UDP	via-ftp	VIA Systems – FTP
64	TCP, UDP	covia	Communications Integrator (CI)
65	TCP, UDP	tacacs-ds	TACACS-Database Service
66	TCP, UDP	sql*net	Oracle SQL.*NET
67	TCP, UDP	bootpc	<u>DHCP/BOOTP Protocol Server</u>
68	TCP, UDP	bootpc	<u>DIICP/BOOTP Protocol Server</u>
69	TCP, UDP	tftp	<u>Trivial File Transfer Protocol</u>
70	TCP, UDP	gopher	<u>Gopher</u>
71	TCP, UDP	netrjs-1	Remote Job Service
72	TCP, UDP	netrjs-2	Remote Job Service
73	TCP, UDP	netrjs-3	Remote Job Service
74	TCP, UDP	netrjs-4	Remote Job Service
75	UDP	T/A	Any private dial-out service
76	TCP, UDP	T/A	Belum ditetapkan

77	TCP, UDP		Any private RJE service
78	TCP, UDP	vetTCP	VetTCP
79	TCP, UDP	finger	Finger
80	TCP, UDP	www	<u>World Wide Web HTTP</u>
81	TCP, UDP	hosts2-ns	HOSTS2 Name Server
82	TCP, UDP	xfer	XFER Utility
83	TCP, UDP	mit-ml-dev	MIT ML Device
84	TCP, UDP	ctf	Common Trace Facility
85	TCP, UDP	mit-ml-dev	MIT ML Device
86	TCP, UDP	mfcobol	Micro Focus Cobol
87	TCP, UDP		Any private terminal link; alias = ttylink
88	TCP, UDP	kerberos	<u>Kerberos</u>
89	TCP, UDP	su-mit-tg	SU/MIT Telnet Gateway
90	TCP, UDP		DNSIX Security Attribute Token Map
91	TCP, UDP	mit-dov	MIT Dover Spooler
92	TCP, UDP	npp	Network Printing Protocol
93	TCP, UDP	dcp	Device Control Protocol
94	TCP, UDP	objcall	Tivoli Object Dispatcher
95	TCP, UDP	supdup	SUPDUP
96	TCP, UDP	dixie	DLXIE Protocol Specification
97	TCP, UDP	swift-rvf	Swift Remote Virtual File Protocol
98	TCP, UDP	tacnews	TAC News
99	TCP, UDP	metagram	Metagram Relay
100	TCP	newacct	(unauthorized use)
101	TCP, UDP	hostname	NIC Host Name Server; alias = hostname
102	TCP, UDP	iso-tsap	ISO-TSAP
103	TCP, UDP	gppitnp	Genesis Point-to-Point Trans Net; alias = webster
104	TCP, UDP	acr-nema	ACR-NEMA Digital Imag. & Comm. 300
105	TCP, UDP	csnet-ns	Mailbox Name Nameserver
106	TCP, UDP	3com-tsmux	3COM-TSMUX
107	TCP, UDP	rtelnet	Remote Telnet Service
108	TCP, UDP	snagas	SNA Gateway Access Server
109	TCP, UDP	pop2	<u>Post Office Protocol version 2 (POP2)</u> ; alias = postoffice
110	TCP, UDP	pop3	<u>Post Office Protocol version 3 (POP3)</u> ; alias = postoffice
111	TCP, UDP	sunrpc	SUN Remote Procedure Call
112	TCP, UDP	mcidas	McIDAS Data Transmission Protocol
113	TCP, UDP	auth	Authentication Service; alias = authentication
114	TCP, UDP	audionews	Audio News Multicast
115	TCP, UDP	sftp	<u>Simple File Transfer Protocol</u>

116	TCP, UDP	ansanotify	ANSA REX Notify
117	TCP, UDP	uucp-path	UUCP Path Service
118	TCP, UDP	sqlserv	SQL Services
119	TCP, UDP	nntp	<u>Network News Transfer Protocol (NNTP)</u> ; alias = usenet
120	TCP, UDP	cfdpkt	CFDPTKT
121	TCP, UDP	erpc	Encore Expedited Remote Procedure Call
122	TCP, UDP	smakynet	SMAKYNET
123	TCP, UDP	ntp	<u>Network Time Protocol</u> ; alias = ntpd ntp
124	TCP, UDP	ansatrader	ANSA REX Trader
125	TCP, UDP	locus-map	Locus PC-Interface Net Map Server
126	TCP, UDP	unitary	Unisys Unitary Login
127	TCP, UDP	locus-con	Locus PC-Interface Connection Server
128	TCP, UDP	gss-xlicen	GSS X License Verification
129	TCP, UDP	pwdgen	Password Generator Protocol
130	TCP, UDP	cisco-fna	Cisco FNATIVE
131	TCP, UDP	cisco-tna	Cisco TNATIVE
132	TCP, UDP	cisco-sys	Cisco SYSMAINT
133	TCP, UDP	statsrv	Statistics Service
134	TCP, UDP	ingres-net	INGRES-NET Service
135	TCP, UDP	loc-srv	Location Service
136	TCP, UDP	profile	PROFILE Naming System
137	TCP, UDP	netbios-ns	<u>NetBIOS Name Service</u>
138	TCP, UDP	netbios-dgm	<u>NetBIOS Datagram Service</u>
139	TCP, UDP	netbios-ssn	<u>NetBIOS Session Service</u>
140	TCP, UDP	emfis-data	EMFIS Data Service
141	TCP, UDP	emfis-ctrl	EMFIS Control Service
142	TCP, UDP	bl-idm	Britton-Lee IDM
143	TCP, UDP	imap2	<u>Interim Mail Access Protocol v2</u>
144	TCP, UDP	news	NewS; alias = news
145	TCP, UDP	uaac	UAAC Protocol
146	TCP, UDP	iso-ip0	ISO-IP0
147	TCP, UDP	iso-ip	ISO-IP
148	TCP, UDP	cronus	CRONUS-SUPPORT
149	TCP, UDP	aed-512	AED 512 Emulation Service
150	TCP, UDP	sql-net	SQL-NET
151	TCP, UDP	hems	HEMS
152	TCP, UDP	bftp	Background File Transfer Program
153	TCP, UDP	sgmp	SGMP; alias = sgmp
154	TCP, UDP	netsc-prod	Netscape
155	TCP, UDP	netsc-dev	Netscape

156	TCP, UDP	sqlsrv	SQL Service
157	TCP, UDP	knet-cmp	KNET/VM Command/Message Protocol
158	TCP, UDP	pcmail-srv	PCMail Server, alias = repository
159	TCP, UDP	nss-routing	NSS-Routing
160	TCP, UDP	sgmp-traps	SGMP-TRAPS
161	TCP, UDP	snmp	<u>Simple Network Management Protocol</u>
162	TCP, UDP	snmptrap	<u>SNMP TRAP</u>
163	TCP, UDP	cmip-man	CMIP/TCP Manager
164	TCP, UDP	cmip-agent	CMIP/TCP Agent
165	TCP, UDP	xns-courier	Xerox
166	TCP, UDP	s-net	Sirius Systems
167	TCP, UDP	namp	NAMP
168	TCP, UDP	rsvd	RSVD
169	TCP, UDP	send	SEND
170	TCP, UDP	print-srv	Network PostScript
171	TCP, UDP	multiplex	Network Innovations Multiplex
172	TCP, UDP	cl/l	Network Innovations CL/1
173	TCP, UDP	xyplex-mux	Xyplex
174	TCP, UDP	mailq	MAILQ
175	TCP, UDP	vmnet	VMNET
176	TCP, UDP	genrad-mux	GENRAD-MUX
177	TCP, UDP	xdmcp	X Display Manager Control Protocol
178	TCP, UDP	nextstep	NextStep Window Server
179	TCP, UDP	bgp	Border Gateway Protocol (BGP)
180	TCP, UDP	ris	Intergraph
181	TCP, UDP	unify	Unify
182	TCP, UDP	audit	Unisys Audit SITP
183	TCP, UDP	ocbinder	OCBinder
184	TCP, UDP	ocserver	OCServer
185	TCP, UDP	remote-kis	Remote-KIS
186	TCP, UDP	kis	KIS Protocol
187	TCP, UDP	aci	Application Communication Interface
188	TCP, UDP	mumps	Plus Five's MUMPS
189	TCP, UDP	qft	Queued File Transport
190	TCP, UDP	gacp	Gateway Access Control Protocol
191	TCP, UDP	prospero	Prospero
192	TCP, UDP	osu-nms	OSU Network Monitoring System
193	TCP, UDP	srmp	Spider Remote Monitoring Protocol
194	TCP, UDP	irc	<u>Internet Relay Chat (IRC) Protocol</u>
195	TCP, UDP	dn6-nlm-aud	DNSIX Network Level Module Audit
196	TCP, UDP	dn6-smmred	DNSIX Session Management Module Audit

			Redirector
197	TCP, UDP	dls	Directory Location Service
198	TCP, UDP	dls-mon	Directory Location Service Monitor
199	TCP, UDP	smux	SMUX
200	TCP, UDP	src	IBM System Resource Controller
201	TCP, UDP	at-rtmp	AppleTalk Routing Maintenance
202	TCP, UDP	at-nbp	<u>AppleTalk</u> Name Binding
203	TCP, UDP	at-3	AppleTalk Unused
204	TCP, UDP	at-echo	AppleTalk Echo
205	TCP, UDP	at-5	AppleTalk Unused
206	TCP, UDP	at-zis	<u>AppleTalk</u> Zone Information
207	TCP, UDP	at-7	<u>AppleTalk</u> Unused
208	TCP, UDP	at-8	<u>AppleTalk</u> Unused
209	TCP, UDP	tam	Trivial Authenticated Mail Protocol
210	TCP, UDP	z39.50	ANSI Z39.50
211	TCP, UDP	914c/g	<u>Texas Instruments</u> 914C/G Terminal
212	TCP, UDP	anet	ATEXSSTR
213	TCP, UDP	ipx	<u>Internetwork Packet Exchange</u> (IPX)
214	TCP, UDP	vmpwscs	VM PWSCS
215	TCP, UDP	softpc	Insignia Solutions
216	TCP, UDP	atls	Access Technology License Server
217	TCP, UDP	dbase	dBASE UNIX
218	TCP, UDP	mpp	Netix Message Posting Protocol
219	TCP, UDP	uarps	Unisys ARPs
220	TCP, UDP	imap3	<u>Interactive Mail Access Protocol versi 3</u>
221	TCP, UDP	fln-spx	Berkeley rlogind with SPX authentication
222	TCP, UDP	fsh-spx	Berkeley rshd with SPX authentication
223	TCP, UDP	cdc	Certificate Distribution Center
224–241	T/A	T/A	Tidak digunakan; dicadangkan
242	TCP, UDP	direct	Direct
243	TCP, UDP	sur-meas	Survey Measurement
245	TCP, UDP	link	LINK
246	TCP, UDP	dsp3270	Display Systems Protocol
247	TCP, UDP	subntbcst_tftp	SUBNTBCST_TFTP
248	TCP, UDP	bhfhs	bhfhs
249–255	T/A	T/A	Tidak digunakan; dicadangkan
345	TCP, UDP	pawserv	Perf Analysis Workbench
346	TCP, UDP	zserv	Zebra server
347	TCP, UDP	faterv	Fatmen Server
371	TCP, UDP	clearcase	Clearcase
372	TCP, UDP	ulistserv	UNIX Listserv

373	TCP, UDP	legent-1	Legent Corporation
374	TCP, UDP	legent-2	Legent Corporation



Artikel mengenai Jaringan komputer ini adalah suatu tulisan rintisan. Anda dapat membantu Wikipedia mengembangkannya.

Diperoleh dari "http://id.wikipedia.org/wiki/Port_TCP_dan_UDP"

```
Procedure TForm_Utama.pertanyaan(status : String);
Var
  xo,a: string;
Begin
  if length(status) = 16 then
  begin
    a := Copy(status,1,4);
    echo := status;
    //if status + eko = eko then lop := echo;
    if length(edit15.Text) = 12 then
    begin
      if a = 'C1BE' then ek := echo;
      if a = 'C2BD' then ko := echo;
      if a = 'C3BC' then oa := echo;
      if a = 'C4BB' then ap := echo;
      if a = 'C5BA' then pr := echo;
      if a = 'C6B9' then ri := echo;
      if a = 'C7B8' then iy := echo;
      if a = 'C8B7' then ya := echo;
      if a = 'C9B6' then an := echo;
      if a = 'CAB5' then nt := echo;
      if a = 'CBB4' then la := echo;
      if a = 'CCB3' then
      begin
        if memo4.Lines.Count >= 2 then Memo4.Lines.Clear;
        memo4.Lines.Add(Copy(status,5,12));
      end;
    end;
  else if length(status) = 16 then //8EF10000007F7F7F
  begin
    echo := status+eko;
    if status = '8EF10000007F7F7F' then tul := echo;
    if apr = Copy(echo,1,16) then ClientSocket1.Socket.SendText(Copy(echo,17,12));
  end;
  if apr = 'C1BE' then ClientSocket1.Socket.SendText(Copy(ek,5,12));
  if apr = 'C2BD' then ClientSocket1.Socket.SendText(Copy(ko,5,12));
  if apr = 'C3BC' then ClientSocket1.Socket.SendText(Copy(oa,5,12));
  if apr = 'C4BB' then ClientSocket1.Socket.SendText(Copy(ap,5,12));
  if apr = 'C5BA' then ClientSocket1.Socket.SendText(Copy(pr,5,12));
  if apr = 'C6B9' then ClientSocket1.Socket.SendText(Copy(ri,5,12));
  if apr = 'C7B8' then ClientSocket1.Socket.SendText(Copy(iy,5,12));
  if apr = 'C8B7' then ClientSocket1.Socket.SendText(Copy(ya,5,12));
  if apr = 'C9B6' then ClientSocket1.Socket.SendText(Copy(an,5,12));
  if apr = 'CAB5' then ClientSocket1.Socket.SendText(Copy(nt,5,12));
  if apr = 'CBB4' then ClientSocket1.Socket.SendText(Copy(la,5,12));
  if apr = 'CCB3' then
  begin
    ClientSocket1.Socket.SendText(copy(Edit16.Text,1,12));
  end;
  if apr = '8EF10000007F7F7F' then ClientSocket1.Socket.SendText(Copy(tul,17,12));
  if apr = '8EF10100007E7F7F' then ClientSocket1.Socket.SendText(Copy(tul,17,12));
  if apr = '8EF10200007D7F7F' then ClientSocket1.Socket.SendText(Copy(tul,17,12));
  apr := "";
```

```
    Timer1.Enabled := true;
  end;
end;

procedure TForm_Utama.VaComm1RxChar(Sender: TObject; Count: Integer);
var
  I: Integer;
  Tmp,terima: string;
begin
  Tmp := VaComm1.ReadText;

  for I := 1 to Length(Tmp) do
    Data_Terima := Data_Terima + DecToHex(ord(Tmp[I]));

  IF Length(Data_Terima) = 12
  Then
    begin
      Edit15.Text := Data_Terima;
      Cek_Terima(Data_Terima);

      Data_Terima := ''; //reset received message
    end;
  end;

procedure TForm_Utama.Timer1Timer(Sender: TObject);
var
  Status_Panel,
  Status_Cahaya : String;
  nomor_Panel : Byte;
begin
  IF RadioGroup1.ItemIndex = 12
  Then
    Begin
      RadioGroup1.ItemIndex := 0;
      Case Cahaya of
        0: Status_Cahaya:= '8EF10000007F7F7F';
        1: Status_Cahaya:= '8EF10100007E7F7F';
        2: Status_Cahaya:= '8EF10200007D7F7F';
        3: Status_Cahaya:= '8EF10000007F7F7F';
      End;
      Kirim(Status_Cahaya);
      PERTANYAAN(Status_Cahaya);
      Edit14.Text := Status_Cahaya;
      IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
      IF CheckBox1.Checked Then memo1.Lines.Add(Status_Cahaya);
      awal := 1;
    End
  Else
    if awal = 0
    then
      RadioGroup1.ItemIndex := RadioGroup1.ItemIndex + 1
    Else
      awal := 0;
    end;
  end;
end;
```

```
IF RadioGroup1.ItemIndex = 12
Then
  Begin
    Kirim('CEBI');
  End
Else
  Begin
    nomor_Panel := RadioGroup1.ItemIndex+193;
    Status_Panel := DecToHex(nomor_Panel);
    Status_Panel := Status_Panel + DecToHex(not(nomor_Panel) OR $80);

    Kirim(Status_Panel);
    PERTANYAAN(Status_Panel);
    Edit14.Text := Status_Panel;
    pertanyaan("");
    IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
    IF CheckBox1.Checked Then memo1.Lines.Add(Status_Panel);

  End;

IF panel_Error[RadioGroup1.ItemIndex+1] < 5
Then
  panel_Error[RadioGroup1.ItemIndex+1] :=
    panel_Error[RadioGroup1.ItemIndex+1] + 1;
end;

procedure TForm1.Utama.ServerSocket1ClientRead(Sender: TObject;
Socket: TCustomWinSocket);
var x,datanya : string;
a,b : integer;
begin
  x:= Socket.ReceiveText;
  //LANTAI 1
  if x ='3,18' then Tombol_1117Click(Tombol_3118); //LAB.SIS TENAGA
  if x ='3,3' then Tombol_1101Click(TOMBOL_3103); //K.LAB.SIS TENAGA
  if x ='3,2' then Tombol_1101Click(Tombol_3102); //RUANG KONTROL
  if x ='3,1' then Tombol_1101Click(Tombol_3101); //GUDANG
  if x ='3,19' then Tombol_1117Click(Tombol_3119); //PERPUSTAKAAN
  if x ='2,6' then Tombol_1101Click(Tombol_2106); //WC 1
  if x ='2,7' then Tombol_1101Click(Tombol_2107); //WC 2
  if x ='2,8' then Tombol_1101Click(Tombol_2108); //WC 3
  if x ='2,5' then Tombol_1101Click(Tombol_2105); //WC 4
  if x ='2,9' then Tombol_1101Click(Tombol_2109); //WC LUAR
  if x ='2,10' then Tombol_1101Click(Tombol_2110); //ELITE
  if x ='1,4' then Tombol_1101Click(Tombol_1104); //R.SEMINAR
  if x ='1,3' then Tombol_1101Click(Tombol_1103); //R.DOSEN
  if x ='1,2' then Tombol_1101Click(Tombol_1102); //R.KORBID
  if x ='1,7' then Tombol_1107Click(Tombol_1107); //SSDP1
  if x ='1,17' then Tombol_1117Click(Tombol_1117); //HALL1
  if x ='1,18' then Tombol_1117Click(Tombol_1118); //HALL2
  if x ='1,19' then Tombol_1117Click(Tombol_1119); //LAB.KONV
  if x ='1,1' then Tombol_1101Click(Tombol_1101); //K.LAB.KONV
  if x ='2,18' then Tombol_1117Click(Tombol_2118); //LAB.TRANSMISI
```

```
if x = '2,1' then Tombol_1101Click(Tombol_2101); //K.LAB.TRANSMISI
if x = '2,19' then Tombol_1117Click(Tombol_2119); //WORKSHOP
if x = '2,2' then Tombol_1101Click(Tombol_2102); //K.WORKSHOP
if x = '1,6' then Tombol_1101Click(Tombol_1106); //TERAS DPN
if x = '3,5' then Tombol_1101Click(Tombol_3105); //TERAS BLKNG
if x = '3,6' then Tombol_1101Click(Tombol_3106); //TERAS KIRI
if x = '2,4' then Tombol_1101Click(Tombol_2104); //TERAS KANAN
if x = '1,5' then Tombol_1101Click(Tombol_1105); //DEPAN KIRI
if x = '2,3' then Tombol_1101Click(Tombol_2103); //DEPAN KANAN
if x = 'e' then Button2.Click;
if x = 'k' then Button3.Click;
if x = 'o' then Button4.Click;
```

//LANTAI DUA

```
if x = '6,19' then Tombol_1117Click(Tombol_3219); //LAB.PENGK
if x = '6,5' then Tombol_1101Click(Tombol_3205); //K.LAB.PENGK
if x = '6,3' then Tombol_1101Click(Tombol_3203); //R.SEKJUR
if x = '6,2' then Tombol_1101Click(Tombol_3202); //R.RECORDING
if x = '6,4' then Tombol_1101Click(Tombol_3204); //R.ADMIN
if x = '6,1' then Tombol_1101Click(Tombol_3201); //R.KAJUR
if x = '5,19' then Tombol_1117Click(Tombol_3118); //LAB.JARKOM
if x = '5,4' then Tombol_1101Click(Tombol_1103); //K.LAB.JARKOM
if x = '5,6' then Tombol_1101Click(Tombol_2206); //WC 1
if x = '5,7' then Tombol_1101Click(Tombol_2207); //WC 2
if x = '5,8' then Tombol_1101Click(Tombol_2208); //WC 3
if x = '5,9' then Tombol_1101Click(Tombol_2209); //WC 4
if x = '5,10' then Tombol_1101Click(Tombol_2210); //WC 5
if x = '5,5' then Tombol_1101Click(Tombol_2205); //R.MAKAN
if x = '5,12' then Tombol_1101Click(Tombol_2212); //R.PANTRI
if x = '4,4' then Tombol_1101Click(Tombol_1204); //HALI. 1
if x = '5,3' then Tombol_1101Click(Tombol_2203); //HALI. 2
if x = '4,3' then Tombol_1101Click(Tombol_1203); //TRS.KIRI
if x = '5,11' then Tombol_1101Click(Tombol_2211); //TRS.KANAN
if x = '4,19' then Tombol_1117Click(Tombol_1219); //LAB.K.INDUSTRI
if x = '4,1' then Tombol_1101Click(Tombol_1201); //K.LAB.K.INDUSTRI
if x = '4,18' then Tombol_1117Click(Tombol_1218); //LAB.PERK.LUNAK
if x = '4,2' then Tombol_1101Click(Tombol_1202); //K.LAB.PERK.LUNAK
if x = '5,17' then Tombol_1117Click(Tombol_2217); //LAB.PEMROG.KOMP
if x = '5,1' then Tombol_1101Click(Tombol_2201); //K.LAB.PEMROG.KOMP
if x = '5,18' then Tombol_1117Click(Tombol_2218); //LAB.ELKA KOMPONEN
if x = '5,2' then Tombol_1101Click(Tombol_2202); //K.LAB.ELKA KOMPONEN
if x = '4,5' then Tombol_1107Click(Tombol_1205); //SSDP 1
if x = '4,6' then Tombol_1107Click(Tombol_1206); //SSDP 2
if x = '6,6' then Tombol_1107Click(Tombol_3206); //SSDP 3
if x = '6,7' then Tombol_1107Click(Tombol_3207); //SSDP 4
```

//LANTAI 3

```
if x = '7,18' then Tombol_1117Click(Tombol_1318); //LAB.ELKA ANALOG
if x = '7,1' then Tombol_1101Click(Tombol_1301); //K.LAB.ELKA ANALOG
if x = '7,17' then Tombol_1117Click(Tombol_1317); //LAB.DSR PEMROG KOMP
if x = '7,5' then Tombol_1101Click(Tombol_1305); //K.LAB.DSR PEMROG KOMP
if x = '7,3' then Tombol_1101Click(Tombol_1303); //TANGGA
```

```
if x = '9,17' then Tombol_1117Click(Tombol_3319); //R.SIM
if x = '9,16' then Tombol_1117Click(Tombol_3316); //R.DIII
if x = '8,8' then Tombol_1101Click(Tombol_2308); //WUDU 1
if x = '8,9' then Tombol_1101Click(Tombol_2309); //WUDU 2
if x = '9,11' then Tombol_1101Click(Tombol_3311); //MUSHOLLA
if x = '9,5' then Tombol_1101Click(Tombol_3305); //WC 1
if x = '9,6' then Tombol_1101Click(Tombol_3306); //WC 2
if x = '9,7' then Tombol_1101Click(Tombol_3307); //WC 3
if x = '9,8' then Tombol_1101Click(Tombol_3308); //WC 4
if x = '7,4' then Tombol_1107Click(Tombol_1304); //SSDP 1
if x = '8,5' then Tombol_1107Click(Tombol_2305); //SSDP 2
if x = '8,6' then Tombol_1107Click(Tombol_2306); //SSDP 3
if x = '8,7' then Tombol_1107Click(Tombol_2307); //SSDP 4
if x = '9,12' then Tombol_1107Click(Tombol_3312); //SSDP 5
if x = '8,4' then Tombol_1101Click(Tombol_2304); //HALL 1
if x = '9,9' then Tombol_1101Click(Tombol_3309); //HALL 2
if x = '8,19' then Tombol_1117Click(Tombol_2319); //LAB.ELKA DIGITAL
if x = '8,3' then Tombol_1101Click(Tombol_2303); //K.LAB.ELKA DIGITAL
if x = '8,18' then Tombol_1117Click(Tombol_2318); //LAB.PEMELIHARAAN
if x = '8,2' then Tombol_1101Click(Tombol_2302); //K.LAB.PEMELIHARAAN
if x = '9,19' then Tombol_1117Click(Tombol_3319); //LAB.MULTIMEDIA
if x = '9,2' then Tombol_1101Click(Tombol_3302); //K.LAB.MULTIMEDIA
if x = '9,1' then Tombol_1101Click(Tombol_3301); //R.KORBID
if x = '9,18' then Tombol_1117Click(Tombol_3318); //RUANG SIDANG 1
if x = '9,10' then Tombol_1101Click(Tombol_3310); //RUANG SIDANG 2
if x = '7,2' then Tombol_1101Click(Tombol_1302); //BELAKANG KIRI
if x = '9,4' then Tombol_1101Click(Tombol_3304); //BELAKANG KANAN
if x = '8,1' then Tombol_1101Click(Tombol_2301); //DEPAN KIRI
if x = '9,3' then Tombol_1101Click(Tombol_3303); //DEPAN KANAN
```

//LANTAI 4

```
if x = '10,17' then Tombol_1117Click(Tombol_1417); //R.SEMINAR
if x = '10,18' then Tombol_1117Click(Tombol_1418); //R.KORBID
if x = '10,19' then Tombol_1117Click(Tombol_1419); //GUDANG
if x = '11,1' then Tombol_1101Click(Tombol_2401); //WC 1
if x = '11,2' then Tombol_1101Click(Tombol_2402); //WC 2
if x = '11,3' then Tombol_1101Click(Tombol_2403); //WC 3
if x = '11,4' then Tombol_1101Click(Tombol_2404); //WC 4
if x = '10,2' then Tombol_1101Click(Tombol_1402); //HALL
if x = '12,19' then Tombol_1117Click(Tombol_3419); //H.M.E
if x = '12,3' then Tombol_1101Click(Tombol_3403); //K.LAB.
if x = '12,18' then Tombol_1117Click(Tombol_3418); //LAB.VIDIO AUDIO
if x = '12,2' then Tombol_1101Click(Tombol_3402); //K.LAB.VIDIO AUDIO
if x = '11,18' then Tombol_1117Click(Tombol_2418); //R.DOSEN
if x = '11,19' then Tombol_1117Click(Tombol_2419); //LAB.TELEKOMUNIKASI
if x = '11,6' then Tombol_1101Click(Tombol_2406); //K.LAB.LAB.TELKOMUNIKASI
if x = '10,1' then Tombol_1101Click(Tombol_1401); //TERAS KIRI BELAKANG
if x = '12,1' then Tombol_1101Click(Tombol_3401); //TERAS KIRI DEPAN
if x = '11,5' then Tombol_1101Click(Tombol_2405); //TERSAS KANAN
if x = '12,4' then Tombol_1107Click(Tombol_3404); //SSDP 1
if x = '12,5' then Tombol_1107Click(Tombol_3405); //SSDP 2
```



```
//minta data
if x='C1BE' then apr := 'C1BE';
if x='C2BD' then apr := 'C2BD';
if x='C3BC' then apr := 'C3BC';
if x='C4BB' then apr := 'C4BB';
if x='C5BA' then apr := 'C5BA';
if x='C6B9' then apr := 'C6B9';
if x='C7B8' then apr := 'C7B8';
if x='C8B7' then apr := 'C8B7';
if x='C9B6' then apr := 'C9B6';
if x='CAB5' then apr := 'CAB5';
if x='CBB4' then apr := 'CBB4';
if x='CCB3' then apr := 'CCB3';
if x='8EF1000007F7F7F' then apr := '8EF1000007F7F7F';
if x='8EF10100007E7F7F' then apr := '8EF10100007E7F7F';
if x='8EF10200007D7F7F' then apr := '8EF10200007D7F7F';
end;

procedure TForm_Utama.ServerSocket1ClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Label31.Caption := 'connected';
  ClientSocket1.Address := Socket.RemoteAddress;
  ClientSocket1.Active := True;
  Label35.Caption := 'Terkoneksi';
  Label35.Font.Color := clGreen;
end;

procedure TForm_Utama.ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Label31.Caption := '';
  Label35.Caption := 'Terputus';
  Label35.Font.Color := clRed;
  ClientSocket1.Active := False;
end;

procedure TForm_Utama.ServerClick(Sender: TObject);
begin
  IF Server.ItemIndex = 0
  Then
    Begin
      ServerSocket1.Active := True;
      Label33.Caption := 'Aktif';
      Label33.Font.Color := clGreen;
    End
  Else
    Begin
      ServerSocket1.Active := False;
      Label33.Caption := 'Tidak Aktif';
      Label33.Font.Color := clRed;
    End;
end;
```



```
procedure TForm_Utama.Edit14Change(Sender: TObject);
var
x: string;
begin
    if Length(Edit14.Text) = 16 then
    begin
        x := Edit14.Text+Edit15.Text;
        Memo3.Lines.Add(x);
        // if CheckBox2.Checked then Memo3.Lines.Add(Edit14.Text+Edit15.Text);
    end;
end;

procedure TForm_Utama.Edit15Change(Sender: TObject);
var
    x: string;
begin
    x := Edit14.Text+Edit15.Text;
    Pertanyaan(x);
    if CheckBox2.Checked then Memo3.Lines.Add(x);
end;

procedure TForm_Utama.Memo4Change(Sender: TObject);
var
    v : string;
begin
    IF Memo4.Lines.Count = 2 Then
    begin
        Edit16.Text := memo4.Lines.CommaText;
        v := memo4.Lines.GetText;
        Memo5.Lines.Clear;
        Memo5.Lines.Add(v);
    end;
end;

procedure TForm_Utama.Button6Click(Sender: TObject);
begin
    Memo4.Lines.Add(Edit16.Text);
end;

procedure TForm_Utama.Edit16Change(Sender: TObject);
var
    x: string;
begin
    if Length(Edit4.Text) = 25 then
    begin
        x := Edit4.Text;
        Memo3.Lines.Add(Copy(x,1,12));
        Memo3.Lines.Add(Copy(x,14,25));
    end;
end;
```

```
procedure TForm_Utama.Edit17Change(Sender: TObject);
```

```
var
```

```
  x,b,c,d,e,f,g,h,i,j,k,l,m: string;
```

```
begin
```

```
  x:= Edit17.Text;
```

```
//1
```

```
  if Copy(x,2,1) = '1' then
```

```
  begin
```

```
    b:= x;
```

```
    Delay(9100);
```

```
    if Copy(b,5,12) <> Copy(ek,5,12) then
```

```
    begin
```

```
      kirim(b);
```

```
      Memo1.Lines.Add(b);
```

```
      Delay(9100);
```

```
      if Copy(b,5,12) <> Copy(ek,5,12) then
```

```
      begin
```

```
        kirim(b);
```

```
        Memo1.Lines.Add(b);
```

```
      end;
```

```
    end;
```

```
  end;
```

```
//2
```

```
  if Copy(x,2,1) = '2' then
```

```
  begin
```

```
    c:= x;
```

```
    Delay(9100);
```

```
    if Copy(c,5,12) <> Copy(ko,5,12) then
```

```
    begin
```

```
      kirim(c);
```

```
      Memo1.Lines.Add(c);
```

```
      Delay(9100);
```

```
      if Copy(c,5,12) <> Copy(ko,5,12) then
```

```
      begin
```

```
        kirim(c);
```

```
        Memo1.Lines.Add(c);
```

```
      end;
```

```
    end;
```

```
  end;
```

```
//3
```

```
  if Copy(x,2,1) = '3' then
```

```
  begin
```

```
    d:= x;
```

```
    Delay(9100);
```

```
    if Copy(d,5,12) <> Copy(oa,5,12) then
```

```
    begin
```

```
      kirim(d);
```

```
      Memo1.Lines.Add(d);
```

```
      Delay(9100);
```

```
      if Copy(d,5,12) <> Copy(oa,5,12) then
```

```
      begin
```

```
        kirim(d);
```

```
        Memo1.Lines.Add(d);
    end;
end;
end;
//4
if Copy(x,2,1) = '4' then
begin
    e:= x;
    Delay(9100);
    if Copy(e,5,12) <> Copy(ap,5,12) then
    begin
        kirim(e);
        Memo1.Lines.Add(e);
        Delay(9100);
        if Copy(e,5,12) <> Copy(ap,5,12) then
        begin
            kirim(e);
            Memo1.Lines.Add(e);
        end;
    end;
end;
//5
if Copy(x,2,1) = '5' then
begin
    f:= x;
    Delay(9100);
    if Copy(f,5,12) <> Copy(pr,5,12) then
    begin
        kirim(f);
        Memo1.Lines.Add(f);
        Delay(9100);
        if Copy(f,5,12) <> Copy(pr,5,12) then
        begin
            kirim(f);
            Memo1.Lines.Add(f);
        end;
    end;
end;
//6
if Copy(x,2,1) = '6' then
begin
    g:= x;
    Delay(9100);
    if Copy(g,5,12) <> Copy(ri,5,12) then
    begin
        kirim(g);
        Memo1.Lines.Add(g);
        Delay(9100);
        if Copy(g,5,12) <> Copy(ri,5,12) then
        begin
            kirim(g);
            Memo1.Lines.Add(g);
        end;
    end;
end;
```

```

    end;
end;
//7
if Copy(x,2,1) = '7' then
begin
    h:= x;
    Delay(9100);
    if Copy(h,5,12) <> Copy(iy,5,12) then
    begin
        kirim(h);
        Memo1.Lines.Add(h);
        Delay(9100);
        if Copy(h,5,12) <> Copy(iy,5,12) then
        begin
            kirim(h);
            Memo1.Lines.Add(h);
        end;
    end;
end;
//8
if Copy(x,2,1) = '8' then
begin
    i:= x;
    Delay(9100);
    if Copy(i,5,12) <> Copy(ya,5,12) then
    begin
        kirim(i);
        Memo1.Lines.Add(i);
        Delay(9100);
        if Copy(i,5,12) <> Copy(ya,5,12) then
        begin
            kirim(i);
            Memo1.Lines.Add(i);
        end;
    end;
end;
//9
if Copy(x,2,1) = '9' then
begin
    j:= x;
    Delay(9100);
    if Copy(j,5,12) <> Copy(an,5,12) then
    begin
        kirim(j);
        Memo1.Lines.Add(j);
        Delay(9100);
        if Copy(j,5,12) <> Copy(an,5,12) then
        begin
            kirim(j);
            Memo1.Lines.Add(j);
        end;
    end;
end;
end;

```

```
//A
if Copy(x,2,1) = 'A' then
begin
  k:= x;
  Delay(9100);
  if Copy(k,5,12) <> Copy(nt,5,12) then
  begin
    kirim(k);
    Memo1.Lines.Add(k);
    Delay(9100);
    if Copy(k,5,12) <> Copy(nt,5,12) then
    begin
      kirim(k);
      Memo1.Lines.Add(k);
    end;
  end;
end;

//B
if Copy(x,2,1) = 'B' then
begin
  l:= x;
  Delay(9100);
  if Copy(l,5,12) <> Copy(la,5,12) then
  begin
    kirim(l);
    Memo1.Lines.Add(l);
    Delay(9100);
    if Copy(l,5,12) <> Copy(la,5,12) then
    begin
      kirim(l);
      Memo1.Lines.Add(l);
    end;
  end;
end;

//C
if Copy(x,2,1) = 'C' then
begin
  m:= x;
  Delay(9100);
  if Copy(m,5,12) <> Copy(Edit16.Text,1,12) then
  begin
    kirim(m);
    Memo1.Lines.Add(m);
    Delay(9100);
    if Copy(m,5,12) <> Copy(Edit16.Text,1,12) then
    begin
      kirim(m);
      Memo1.Lines.Add(m);
    end;
  end;
end;
end;
end;
```

Program Pada *Slave Computer*

```
unit Utama;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, ADODB, StdCtrls, Buttons, ExtCtrls, Menus, ComCtrls,  
VaClasses, VaComm, jpeg, Grids, DBGrids, ImgList, ToolWin, Gauges,  
ScktComp;
```

```
type
```

```
TForm Utama = class(TForm)  
  Panel2: TPanel;  
  Bevel2: TBevel;  
  Label3: TLabel;  
  Label4: TLabel;  
  Label5: TLabel;  
  Label6: TLabel;  
  Label7: TLabel;  
  Edit3: TEdit;  
  Edit4: TEdit;  
  Edit5: TEdit;  
  Edit6: TEdit;  
  BitBtn3: TBitBtn;  
  BitBtn4: TBitBtn;  
  Panel1: TPanel;  
  Bevel1: TBevel;  
  Label1: TLabel;  
  Label2: TLabel;  
  Label8: TLabel;  
  Edit1: TEdit;  
  Edit2: TEdit;  
  BitBtn1: TBitBtn;  
  BitBtn2: TBitBtn;  
  VaComm1: TVaComm;  
  Timer1: TTimer;  
  Timer2: TTimer;  
  Timer3: TTimer;  
  Timer4: TTimer;  
  PopupMenu1: TPopupMenu;  
  SetTimer1: TMenuItem;  
  HapusTimer1: TMenuItem;  
  Label18: TLabel;  
  ADOConnection1: TADOConnection;  
  DataSource1: TDataSource;  
  DataSource2: TDataSource;
```

ADOTable2: TADOTable;	Label26: TLabel;
ADOTable1: TADOTable;	Label27: TLabel;
ADOTable1Lantai: TSmallintField;	Label28: TLabel;
ADOTable1Nomor: TIntegerField;	Bevel5: TBevel;
ADOTable1Nama: TWideStringField;	RadioGroup1: TRadioGroup;
ADOTable1Hidup: TDateTimeField;	CheckBox1: TCheckBox;
ADOTable1Mati: TDateTimeField;	Button1: TButton;
ADOTable2Cahaya: TSmallintField;	Memo1: TMemo;
ADOTable2Keterangan:	DBGrid1: TDBGrid;
TwideStringField;	DBGrid2: TDBGrid;
ADOTable2Batas_Bawah:	BitBtn7: TBitBtn;
TSmallintField;	Edit10: TEdit;
Panel3: TPanel;	Edit11: TEdit;
Label13: TLabel;	Edit12: TEdit;
Label14: TLabel;	DateTimePicker3: TDateTimePicker;
Label15: TLabel;	DateTimePicker4: TDateTimePicker;
Label16: TLabel;	BitBtn8: TBitBtn;
Label17: TLabel;	BitBtn9: TBitBtn;
Label19: TLabel;	TabSheet2: TTabSheet;
DateTimePicker1: TDateTimePicker;	Lantai1: TImage;
DateTimePicker2: TDateTimePicker;	Lampu_3106b: TImage;
Edit7: TEdit;	Lampu_3103: TImage;
Edit8: TEdit;	TL_Off_Off: TImage;
Edit9: TEdit;	TL_On_Off: TImage;
BitBtn5: TBitBtn;	TL_On_On: TImage;
BitBtn6: TBitBtn;	Lampu_3118: TImage;
Label20: TLabel;	Lampu_3102: TImage;
Bevel4: TBevel;	Lampu_3101: TImage;
DialogBukaFile: TOpenDialog;	Lampu_3119: TImage;
ImageList1: TImageList;	Lampu_3104b: TImage;
DialogSimpanFile: TSaveDialog;	Lampu_2106: TImage;
OpenDialog1: TOpenDialog;	Lampu_2107: TImage;
OpenDialog2: TOpenDialog;	Lampu_2108: TImage;
ClientSocket1: TClientSocket;	Lampu_2105a: TImage;
ServerSocket1: TServerSocket;	Lampu_2105b: TImage;
PageControl1: TPageControl;	Lampu_2110: TImage;
TabSheet6: TTabSheet;	Lampu_1104: TImage;
Label36: TLabel;	Lampu_1103: TImage;
ToolBar1: TToolBar;	Lampu_1102: TImage;
TabSheet1: TTabSheet;	Lampu_1101: TImage;
Bevel3: TBevel;	Lampu_2101: TImage;
Label21: TLabel;	Lampu_2102: TImage;
Label24: TLabel;	Lampu_2118: TImage;
Label22: TLabel;	Lampu_2119: TImage;
Label23: TLabel;	Lampu_1119: TImage;
Label25: TLabel;	Lampu_3106a: TImage;

Lampu_1202: TImage;
 Lampu_1201: TImage;
 Lampu_2201: TImage;
 Lampu_2202: TImage;
 Lampu_1218: TImage;
 Lampu_3219: TImage;
 Lampu_2217: TImage;
 Lampu_2218: TImage;
 Lampu_2219: TImage;
 Lampu_2206: TImage;
 Lampu_2207: TImage;
 Lampu_2208: TImage;
 Lampu_2209: TImage;
 Lampu_2210: TImage;
 Lampu_2212: TImage;
 Lampu_1206: TImage;
 Lampu_3206: TImage;
 Lampu_3207: TImage;
 Image2: TImage;
 Image6: TImage;
 Panel_12: TImage;
 Panel_22: TImage;
 Panel_32: TImage;
 Lampu_1205: TImage;
 Label10: TLabel;
 Tombol_3203: TPanel;
 Tombol_3202: TPanel;
 Tombol_2217: TPanel;
 Tombol_3205: TPanel;
 Tombol_3204: TPanel;
 Tombol_3201: TPanel;
 Tombol_2206: TPanel;
 Tombol_2207: TPanel;
 Tombol_2208: TPanel;
 Tombol_2210: TPanel;
 Tombol_2209: TPanel;
 Tombol_2218: TPanel;
 Tombol_1204: TPanel;
 Tombol_2203: TPanel;
 Tombol_1218: TPanel;
 Tombol_2219: TPanel;
 Tombol_3219: TPanel;
 Tombol_2205: TPanel;
 Tombol_2212: TPanel;
 Tombol_2204: TPanel;
 Tombol_1202: TPanel;
 Tombol_2201: TPanel;
 Tombol_2202: TPanel;
 Tombol_1205: TPanel;
 Tombol_1206: TPanel;
 Tombol_3207: TPanel;
 Tombol_3206: TPanel;
 Tombol_1201: TPanel;
 Tombol_1219: TPanel;
 Tombol_2211: TPanel;
 Tombol_1203: TPanel;
 Edit13: TEdit;
 Button5: TButton;
 Memo2: TMemo;
 TabSheet4: TTabSheet;
 Lantai3: TImage;
 Lampu_1302: TImage;
 Lampu_1318: TImage;
 Lampu_1301: TImage;
 Lampu_2304a: TImage;
 Lampu_3305: TImage;
 Lampu_1304: TImage;
 Lampu_2301: TImage;
 Lampu_3304: TImage;
 Lampu_3303: TImage;
 Lampu_2304b: TImage;
 Lampu_1317: TImage;
 Lampu_2319: TImage;
 Lampu_2318: TImage;
 Lampu_3316: TImage;
 Lampu_3317: TImage;
 Lampu_3319: TImage;
 Lampu_3318: TImage;
 Lampu_2303: TImage;
 Lampu_2302: TImage;
 Lampu_1305: TImage;
 Lampu_3311: TImage;
 Lampu_3302: TImage;
 Lampu_3301: TImage;
 Lampu_3306: TImage;
 Lampu_3307: TImage;
 Lampu_3308a: TImage;
 Lampu_3308b: TImage;
 Lampu_1303a: TImage;
 Lampu_1303b: TImage;
 Lampu_2304c: TImage;
 Lampu_2304d: TImage;


```

procedure BitBtn7Click(Sender: TObject);
procedure Edit10KeyPress(Sender: TObject; var Key: Char);
procedure DBGrid2CellClick(Column: TColumn);
procedure DBGrid1CellClick(Column: TColumn);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);

```

```

procedure ButtonLoadHurufClick(Sender: TObject);
procedure TombolPetunjukClick(Sender: TObject);
procedure tulis(dd,ee,ff:string);
procedure langsung(dd,ff:string);
procedure DirectDemoClick(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
procedure Button5Click(Sender: TObject);
procedure ServerSocket1ClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ClientClick(Sender: TObject);

```

private

```
{ Private declarations }
```

public

```

Procedure Delay(Lama : Integer);
Procedure Kirim(datanya : String);
Procedure Cek_Terima(datanya : String);
procedure Satu_Relay(Nomor_Panel : byte; Nomor : Integer);
procedure Dua_Relay(Nomor_Panel : byte; Nomor : Integer);
procedure Timer_Satu_Relay(Nomor_Panel : byte; Nomor : Integer; Status : Integer);
procedure Timer_Dua_Relay(Nomor_Panel : byte; Nomor : Integer; Status : Integer);
Procedure Gambar_Lampu(Nomor_Panel: Integer; Data : Integer; Status : Byte);
{ Public declarations }

```

```

Procedure Kirim_Data(kar : byte);
procedure kirim_char(Besar: Boolean; Display: Byte);
procedure tulis_ke_memory_cepat1(Display: byte; List: string);
procedure kesalahan(s: string);

```

end;

Type

```

TData = array [1..1032,1..2] of byte;
FData = file of TData;
Urut = array [0..(515+6)] of byte;

```

var

```

Form_Utama      : TForm_Utama;
Servernya, Usernya,

```

```

Passwordnya,DatabaseNya : String;
Direktorinya      : String;
Operatornya       : String;
Waktu_Log_Off     : TTime;
Proses_Kirim,
Display_Error     : Boolean;
Data_Terima       : String;
D1_Panel,D2_Panel,
D3_Panel          : Array [1..50] of Byte;
// Status         : Array [1..1000] of integer;
panel             : Array [1..44] of integer;
jumlah_Lampu      : Array [100..1500] of Integer;
panel_Error       : Array [1..20] of integer;
Cahaya           : Integer;
Mati,Hidup        : Array [1..100] of TTime;
B_Terang,
B_Sedang          : Integer;

Panjang : Integer;
data_ : URUT;
Cs_IP_Lama      : Pointer;
Huruf, Scan,
Buffer Rx,
CTS, THRE, TSRE,
Check_data_masuk,
Data_masuk,dt,hasil, CDisplayKe,
message_length,TX_data,SEQ : Byte;
y_jkl,Panjang_no_telp,
Panjang_no_center,x,K,
l, m, i, hitung_kirim : Integer;
hitung,Ambil        : longint;
Tx_Buffer           : Word;
Rx_Buffer           : Word;
Baud_Div_LSB        : Word;
Baud_Div_MSB        : Word;
Int_Enb_Reg         : Word;
Int_ID_Reg          : Word;
Line_Ctrl_Reg       : Word;
Mod_Ctrl_Reg        : Word;
Line_Stat_Reg       : Word;
Mod_Stat_Reg        : Word;
jam_kirim           : string;
kar                 : char;
TExt_data           : STRING;
dataGambar          : TData;
dataHuruf           : TData;
f                   : FData;

Awal                : Integer;

```

```
const
```

```

DATA_Char : Urut =
(0,$1F,$24,$44,$24,$1F,{A} 0,$7F,$49,$49,$49,$36,{B} 0,$3E,$41,$41,$41,$22,{C}
0,$7F,$41,$41,$41,$3E,{D} 0,$7F,$49,$49,$49,$41,{E} 0,$7F,$48,$48,$48,$40,{F}
0,$3E,$41,$41,$45,$47,{G} 0,$7F,$08,$08,$08,$7F,{H} 0,$00,$41,$7F,$41,$00,{I}
0,$02,$01,$01,$01,$7E,{J} 0,$7F,$08,$14,$22,$41,{K} 0,$7F,$01,$01,$01,$01,{L}
0,$7F,$20,$18,$20,$7F,{M} 0,$7F,$10,$08,$04,$7F,{N} 0,$3E,$41,$41,$41,$3E,{O}
0,$7F,$48,$48,$48,$30,{P} 0,$3E,$41,$45,$42,$3D,{Q} 0,$7F,$48,$4C,$4A,$31,{R}
0,$32,$49,$49,$49,$26,{S} 0,$40,$40,$7E,$40,$40,{T} 0,$7E,$01,$01,$01,$7E,{U}
0,$7C,$02,$01,$02,$7C,{V} 0,$7F,$02,$0C,$02,$7F,{W} 0,$63,$14,$08,$14,$63,{X}
0,$60,$10,$0F,$10,$60,{Y} 0,$43,$45,$49,$51,$61,{Z} 0,$3E,$45,$49,$51,$3E,{0}
0,0,$21,$7F,$01,$00,{1} 0,$23,$45,$49,$49,$31,{2} 0,$42,$41,$49,$59,$66,{3}
0,$0C,$14,$24,$7F,$04,{4} 0,$72,$51,$51,$51,$4E,{5} 0,$1E,$29,$49,$49,$46,{6}
0,$40,$47,$48,$50,$60,{7} 0,$36,$49,$49,$49,$36,{8} 0,$31,$49,$49,$4A,$3C,{9}
0,0,0,0,0,0,{SPACE} 0,$36,$49,$35,$02,$05,{& 37}
0,$22,$14,$7F,$14,$22,{* 38} 0,0,0,$7D,0,{! 39}
0,0,2,4,8,$10,$20,{/ 40} 0,$14,$14,$14,$14,$14,{= 41}
0,0,0,1,0,0,{. 42} 0,0,0,$14,0,0,{: 43}
0,$7F,$7F,$41,$41,$41,{[ 44} 0,$41,$41,$41,$7F,$7F,{] 45}
0,$04,$08,$10,$08,$04,{^ 46} 0,$14,$7F,$14,$7F,$14,{# 47}
0,$12,$2A,$7F,$2A,$24,{dolar 48} 0,$00,$00,$70,$00,$00,{‘ 49}
0,$1C,$22,$41,$0,$0,{( 50} 0,$0,$0,$41,$22,$1C,{) 51}
0,$08,$08,$3E,$08,$08,{+ 52} 0,$20,$40,$4D,$50,$20,{? 53}
0,$3E,$41,$5D,$4D,$39,{@ 54} 0,$62,$64,$08,$13,$23,{ 55}
0,0,1,6,0,0,{,} 0,0,1,6,0,0,{~}
0,0,1,6,0,0,{'} 0,0,1,6,0,0,{-}
0,0,1,6,0,0,{-}
0,$0E,$11,$11,$1F,$00,{a} 0,$7E,$11,$11,$0E,$00,
0,$0E,$11,$11,$0A,$00, 0,$0E,$11,$11,$7E,$00,
0,$0E,$15,$15,$09,$00, 0,$08,$3F,$48,$20,$00,
0,$19,$25,$25,$1e,$00, 0,$7F,$10,$10,$0F,$00,
0,$10,$5E,$01,$00,$00, 0,$02,$01,$01,$5E,$00,
0,$7F,$0A,$11,$11,$00, 0,$20,$7E,$01,$02,$00,
0,$0F,$10,$0F,$10,$0F, 0,$0F,$10,$10,$0F,$00,
0,$0E,$11,$11,$0E,$00, 0,$1F,$24,$24,$18,$00,
0,$18,$24,$24,$1F,$01, 0,$0F,$10,$08,$00,$00,
0,$09,$15,$15,$12,$00, 0,$10,$7E,$11,$02,$00,
0,$1E,$01,$01,$1E,$00, 0,$1C,$02,$01,$02,$1C,
0,$1E,$01,$1E,$01,$1E, 0,$1b,$04,$04,$1b,$00,
0,$19,$05,$05,$1E,$00, 0,$11,$13,$15,$19,$11); {z}

```

implementation

```
{ $R *.dfm }
```

```
Procedure TForm_Utama.Delay(Lama : Integer);
```

```
var
```

```
Ref : Longint;
```

```
Begin
```

```
Ref := GetTickCount;
```

```
Repeat
```

```
Application.ProcessMessages;
```

```

Until ((GetTickCount - Ref) >= Lama);
End;

```

```

Procedure TForm Utama.Kirim_Data(kar : byte);
Begin
  VaComm1.WriteChar(Chr(kar));
  delay(5);
end;
{ Added at 28 Jan 2004 }

```

```

procedure TForm Utama.kirim_char(Besar:Boolean; Display:Byte);
var
  dat : byte;
  k : Integer;
begin
  dat := ord(kar)-65;
  if kar in ['0'..'9'] then
    dat := ord(kar)-22;
  if kar in ['a'..'z'] then
    dat := dat + 28;
  if kar = '.' then dat := 42;
  if kar = ':' then dat := 43;
  if kar = '=' then dat := 41;
  if kar = '*' then dat := 38;
  if kar = '&' then dat := 37;
  if kar = '!' then dat := 39;
  if kar = '/' then dat := 40;
  if kar = '[' then dat := 44;

  if kar = ']' then dat := 45;
  if kar = '"' then dat := 36;
  if kar = '^' then dat := 46;
  if kar = '#' then dat := 47;
  if kar = '$' then dat := 48;
  if kar = char(039) then dat := 49;
  if kar = '(' then dat := 50;
  if kar = ')' then dat := 51;
  if kar = '+' then dat := 52;
  if kar = '?' then dat := 53;
  if kar = '@' then dat := 54;
  if kar = '%' then dat := 55;
  if kar = ',' then dat := 56;
  if kar = '~' then dat := 57;
  if kar = '"' then dat := 58;
  if kar = ';' then dat := 59;
  if kar = '-' then dat := 60;
  If not Besar then
    for k := (dat * 6) to ((dat * 6) + 5) do
      begin
        Data_masuk := data_Char[k];
        kirim_data(Data_masuk);delay(1);
      end
    end
  end

```

```

else
  for k := (dat * 12) to ((dat * 12) + 11) do
  begin
    Data_masuk := dataHuruf[k,Display];
    kirim_data(Data_masuk);
    delay(1);
  end;
end;

procedure TForm_Utama.tulis_ke_memory_cepat1(Display:byte;List:string);
var hit : integer;
    Tulisan, kode : string;
    tanda : boolean;
    bukan_karakter : boolean;
    selesai : boolean;
    Ambilline : Longint;
Begin
  tanda := false;
  selesai := false;
  kode := '';
  Ambil := 1;
  Ambilline := 1;
  hitung_kirim := 0;
  Tulisan := copy(List,Ambilline,1);
  While(kode <> '<END>') Do
  Begin
    Hasil := ord(Tulisan[Ambil]);
    if tanda = true then
    begin
      kode := kode + char(Hasil);
      inc(hit);
    end;
    if Hasil = ord('<') then
    begin
      hit := 1;
      kode := '<';
      tanda := true;
    end;
    if tanda = false then
    begin
      if hasil <> $0D then
      if hasil <> $0A then
      begin
        kar := char(hasil) ;
        case Display of
          8: kirim_char(True,1); {atas }
          9: kirim_char(True,2); {bawah}
        end;
      end;
    end;
  end;
  if Hasil = ord('>') then
  begin

```

```

tanda := false;
if kode = '<GBR>' then
begin
  case Display of
    8:for k := 1 to 190 do {240}
      begin
        Data_masuk := dataGambar[k,1];
        kirim_data(Data_masuk);
        //delay(1);
      end;
    9:for k := 1 to 190 do {240}
      begin
        Data_masuk := dataGambar[k,2];
        kirim_data(Data_masuk);
        // delay(1);
      end;
  end;
end;
end;
If Ambil = Length(Tulisan) then
Begin
  Ambil := 1;
  inc(AmbilLine);
  if AmbilLine = length(List)+1 then
    break;
  Tulisan := List[AmbilLine];
end
else
  Inc(Ambil);
end;
End;

procedure TForm_Utama.kesalahan(s:string);
begin
  MessageBox (0,pchar(s),'Perhatian',MB_OK or MB_ICONHAND);
end;

procedure TForm_Utama.ButtonLoadHurufClick(Sender: TObject);
begin
  if OpenFileDialog2.Execute then
  begin
    OpenFileDialog2.Title := 'Buka File Huruf',
    if not FileExists(OpenFileDialog2.FileName) then
      Begin
        kesalahan('File Huruf Tidak Ditemukan!');
        exit;
      end;
    AssignFile(f,OpenFileDialog2.FileName);
    Reset(f);
    {SI-}
    seek(f,0);
  end;
end;

```

```

    read(f,dataHuruf);
  {$I+}
  if IOResult <> 0 then
    Begin
      kesalahan('Bukan file peraga!');
      exit;
    end;
  closefile(f);
end;
end;

procedure TForm_Utama.TombolPetunjukClick(Sender: TObject);
begin
  //Petunjuk.Visible := True;
end;

procedure TForm_Utama.tulis(dd,ee,ff:string);
var meja:integer;
    bank:byte;
begin
  meja:=strtoint(dd);
  if ee='1' then bank:=$E0;
  if ee='2' then bank:=$E1;
  if ee='3' then bank:=$E2;
  if ee='4' then bank:=$E3;
  {tulis_ke_memory (meja,bank,ff); }
end;

procedure TForm_Utama.langsung(dd,ff:string);
var meja:integer;
begin
  meja:= StrToInt(dd);
  meja := 8;
  TX_data := meja or $60;
  TX_data := $86 { not(meja) };
  Tulis_ke_memory_cepat1(meja,ff);
  TX_data := (meja + 1) or $60;
  TX_data := $6A { not(CDisplayKe + 1) };
  Tulis_ke_memory_cepat1(meja + 1,ff);
end;

procedure TForm_Utama.DirectDemoClick(Sender: TObject);
begin

end;

```

```
// =====  
//      Program BAS  
// =====
```

```
Function Hexbin(Kar : String) : Integer;  
var  
    Hasil : Integer;  
Begin  
    Hasil := $00;  
  
    IF Kar[1] = '0' Then Hasil := $00  
    Else IF Kar[1] = '1' Then Hasil := $10  
    Else IF Kar[1] = '2' Then Hasil := $20  
    Else IF Kar[1] = '3' Then Hasil := $30  
    Else IF Kar[1] = '4' Then Hasil := $40  
    Else IF Kar[1] = '5' Then Hasil := $50  
    Else IF Kar[1] = '6' Then Hasil := $60  
    Else IF Kar[1] = '7' Then Hasil := $70  
    Else IF Kar[1] = '8' Then Hasil := $80  
    Else IF Kar[1] = '9' Then Hasil := $90  
    Else IF Kar[1] = 'A' Then Hasil := $A0  
    Else IF Kar[1] = 'B' Then Hasil := $B0  
    Else IF Kar[1] = 'C' Then Hasil := $C0  
    Else IF Kar[1] = 'D' Then Hasil := $D0  
    Else IF Kar[1] = 'E' Then Hasil := $E0  
    Else IF Kar[1] = 'F' Then Hasil := $F0;  
  
    IF Kar[2] = '0' Then Hasil := Hasil + $00  
    Else IF Kar[2] = '1' Then Hasil := Hasil + $01  
    Else IF Kar[2] = '2' Then Hasil := Hasil + $02  
    Else IF Kar[2] = '3' Then Hasil := Hasil + $03  
    Else IF Kar[2] = '4' Then Hasil := Hasil + $04  
    Else IF Kar[2] = '5' Then Hasil := Hasil + $05  
    Else IF Kar[2] = '6' Then Hasil := Hasil + $06  
    Else IF Kar[2] = '7' Then Hasil := Hasil + $07  
    Else IF Kar[2] = '8' Then Hasil := Hasil + $08  
    Else IF Kar[2] = '9' Then Hasil := Hasil + $09  
    Else IF Kar[2] = 'A' Then Hasil := Hasil + $0A  
    Else IF Kar[2] = 'B' Then Hasil := Hasil + $0B  
    Else IF Kar[2] = 'C' Then Hasil := Hasil + $0C  
    Else IF Kar[2] = 'D' Then Hasil := Hasil + $0D  
    Else IF Kar[2] = 'E' Then Hasil := Hasil + $0E  
    Else IF Kar[2] = 'F' Then Hasil := Hasil + $0F;  
  
    Result := Hasil;  
End;  
  
Function DecToHex ( Datanya : Integer ) : String;  
Var  
    B : integer;  
const
```



```

A : array [0..15] of Char =
  ('0','1','2','3','4','5','6','7','8',
   '9','A','B','C','D','E','F');
begin
  B := Datanya;
  Result :=
    A[StrToInt(FloatToStr((B - (B mod 16) )/16))] +
    A[B mod 16];
End;

Procedure TForm_Utama.Kirim(datanya : String);
Var
  N : Integer;
Begin
  If not Proses_Kirim
  Then
    Begin
      Proses_Kirim := True;
      For N := 1 To StrToInt(FloatToStr(Length(datanya)/2)) Do
        Begin
          // Form_Utama.VaComm1.WriteChar(Char(Hexbin(Copy(datanya,2*N-1,2))));
          Delay(50);
        End;
      Proses_Kirim := False;
    End;
End;

Function Lampu(nomor : integer) : TImage;
Begin
  Case Nomor of
    101 : Result := Form_Utama.Lampu_1101;
    102 : Result := Form_Utama.Lampu_1102;
    103 : Result := Form_Utama.Lampu_1103;
    104 : Result := Form_Utama.Lampu_1104;
    1051 : Result := Form_Utama.Lampu_1105a;
    1052 : Result := Form_Utama.Lampu_1105b;
    1061 : Result := Form_Utama.Lampu_1106a;
    1062 : Result := Form_Utama.Lampu_1106b;
    1063 : Result := Form_Utama.Lampu_1106c;
    107 : Result := Form_Utama.Lampu_1107;
    1171 : Result := Form_Utama.Lampu_1117a;
    1172 : Result := Form_Utama.Lampu_1117b;
    1173 : Result := Form_Utama.Lampu_1117c;
    1174 : Result := Form_Utama.Lampu_1117d;
    1175 : Result := Form_Utama.Lampu_1117e;
    1176 : Result := Form_Utama.Lampu_1117f;
    1177 : Result := Form_Utama.Lampu_1117g;
    1178 : Result := Form_Utama.Lampu_1117h;
    1181 : Result := Form_Utama.Lampu_1118a;
    1182 : Result := Form_Utama.Lampu_1118b;
    1183 : Result := Form_Utama.Lampu_1118c;
    1184 : Result := Form_Utama.Lampu_1118d;
  end;
End;

```

1185 : Result := Form_Utama.Lampu_1118e;
1186 : Result := Form_Utama.Lampu_1118f;
1187 : Result := Form_Utama.Lampu_1118g;
1188 : Result := Form_Utama.Lampu_1118h;
119 : Result := Form_Utama.Lampu_1119;

201 : Result := Form_Utama.Lampu_2101;
202 : Result := Form_Utama.Lampu_2102;
2031 : Result := Form_Utama.Lampu_2103a;
2032 : Result := Form_Utama.Lampu_2103b;
2033 : Result := Form_Utama.Lampu_2103c;
204 : Result := Form_Utama.Lampu_2104;
2051 : Result := Form_Utama.Lampu_2105a;
2052 : Result := Form_Utama.Lampu_2105b;
206 : Result := Form_Utama.Lampu_2106;
207 : Result := Form_Utama.Lampu_2107;
208 : Result := Form_Utama.Lampu_2108;
209 : Result := Form_Utama.Lampu_2109;
210 : Result := Form_Utama.Lampu_2110;
218 : Result := Form_Utama.Lampu_2118;
219 : Result := Form_Utama.Lampu_2119;

301 : Result := Form_Utama.Lampu_3101;
302 : Result := Form_Utama.Lampu_3102;
303 : Result := Form_Utama.Lampu_3103;
3041 : Result := Form_Utama.Lampu_3104a;
3042 : Result := Form_Utama.Lampu_3104b;
305 : Result := Form_Utama.Lampu_3105;
3061 : Result := Form_Utama.Lampu_3106a;
3062 : Result := Form_Utama.Lampu_3106b;
318 : Result := Form_Utama.Lampu_3118;
319 : Result := Form_Utama.Lampu_3119;

401 : Result := Form_Utama.Lampu_1201;
402 : Result := Form_Utama.Lampu_1202;
4031 : Result := Form_Utama.Lampu_1203a;
4032 : Result := Form_Utama.Lampu_1203b;
4041 : Result := Form_Utama.Lampu_1204a;
4042 : Result := Form_Utama.Lampu_1204b;
4043 : Result := Form_Utama.Lampu_1204c;
4044 : Result := Form_Utama.Lampu_1204d;
4045 : Result := Form_Utama.Lampu_1204e;
405 : Result := Form_Utama.Lampu_1205;
406 : Result := Form_Utama.Lampu_1206;
418 : Result := Form_Utama.Lampu_1218;
419 : Result := Form_Utama.Lampu_1219;

501 : Result := Form_Utama.Lampu_2201;
502 : Result := Form_Utama.Lampu_2202;
5031 : Result := Form_Utama.Lampu_2203a;
5032 : Result := Form_Utama.Lampu_2203b;
5033 : Result := Form_Utama.Lampu_2203c;

5034 : Result := Form_Utama.Lampu_2203d;
 504 : Result := Form_Utama.Lampu_2204;
 505 : Result := Form_Utama.Lampu_2205;
 506 : Result := Form_Utama.Lampu_2206;
 507 : Result := Form_Utama.Lampu_2207;
 508 : Result := Form_Utama.Lampu_2208;
 509 : Result := Form_Utama.Lampu_2209;
 510 : Result := Form_Utama.Lampu_2210;
 5111 : Result := Form_Utama.Lampu_2211a;
 5112 : Result := Form_Utama.Lampu_2211b;
 512 : Result := Form_Utama.Lampu_2212;
 517 : Result := Form_Utama.Lampu_2217;
 518 : Result := Form_Utama.Lampu_2218;
 519 : Result := Form_Utama.Lampu_2219;

601 : Result := Form_Utama.Lampu_3201;
 602 : Result := Form_Utama.Lampu_3202;
 603 : Result := Form_Utama.Lampu_3203;
 604 : Result := Form_Utama.Lampu_3204;
 605 : Result := Form_Utama.Lampu_3205;
 606 : Result := Form_Utama.Lampu_3206;
 607 : Result := Form_Utama.Lampu_3207;
 619 : Result := Form_Utama.Lampu_3219;

701 : Result := Form_Utama.Lampu_1301;
 702 : Result := Form_Utama.Lampu_1302;
 7031 : Result := Form_Utama.Lampu_1303a;
 7032 : Result := Form_Utama.Lampu_1303b;
 704 : Result := Form_Utama.Lampu_1304;
 705 : Result := Form_Utama.Lampu_1305;
 718 : Result := Form_Utama.Lampu_1318;
 717 : Result := Form_Utama.Lampu_1317;

801 : Result := Form_Utama.Lampu_2301;
 802 : Result := Form_Utama.Lampu_2302;
 803 : Result := Form_Utama.Lampu_2303;
 8041 : Result := Form_Utama.Lampu_2304a;
 8042 : Result := Form_Utama.Lampu_2304b;
 8043 : Result := Form_Utama.Lampu_2304c;
 8044 : Result := Form_Utama.Lampu_2304d;
 805 : Result := Form_Utama.Lampu_2305;
 806 : Result := Form_Utama.Lampu_2306;
 807 : Result := Form_Utama.Lampu_2307;
 808 : Result := Form_Utama.Lampu_2308;
 809 : Result := Form_Utama.Lampu_2309;
 818 : Result := Form_Utama.Lampu_2318;
 819 : Result := Form_Utama.Lampu_2319;

901 : Result := Form_Utama.Lampu_3301;
 902 : Result := Form_Utama.Lampu_3302;
 903 : Result := Form_Utama.Lampu_3303;
 904 : Result := Form_Utama.Lampu_3304;

905 : Result := Form_Utama.Lampu_3305;
 906 : Result := Form_Utama.Lampu_3306;
 907 : Result := Form_Utama.Lampu_3307;
 9081 : Result := Form_Utama.Lampu_3308a;
 9082 : Result := Form_Utama.Lampu_3308b;
 9091 : Result := Form_Utama.Lampu_3309a;
 9092 : Result := Form_Utama.Lampu_3309b;
 9093 : Result := Form_Utama.Lampu_3309c;
 9101 : Result := Form_Utama.Lampu_3310a;
 9102 : Result := Form_Utama.Lampu_3310b;
 9103 : Result := Form_Utama.Lampu_3310c;
 9104 : Result := Form_Utama.Lampu_3310d;
 911 : Result := Form_Utama.Lampu_3311;
 912 : Result := Form_Utama.Lampu_3312;
 916 : Result := Form_Utama.Lampu_3316;
 917 : Result := Form_Utama.Lampu_3317;
 918 : Result := Form_Utama.Lampu_3318;
 919 : Result := Form_Utama.Lampu_3319;

 10011 : Result := Form_Utama.Lampu_1401a;
 10012 : Result := Form_Utama.Lampu_1401b;
 10013 : Result := Form_Utama.Lampu_1401c;
 10021 : Result := Form_Utama.Lampu_1402a;
 10022 : Result := Form_Utama.Lampu_1402b;
 10023 : Result := Form_Utama.Lampu_1402c;
 10024 : Result := Form_Utama.Lampu_1402d;
 10025 : Result := Form_Utama.Lampu_1402e;
 1017 : Result := Form_Utama.Lampu_1417;
 1018 : Result := Form_Utama.Lampu_1418;
 1019 : Result := Form_Utama.Lampu_1419;

 1101 : Result := Form_Utama.Lampu_2401;
 1102 : Result := Form_Utama.Lampu_2402;
 1103 : Result := Form_Utama.Lampu_2403;
 11041 : Result := Form_Utama.Lampu_2404a;
 11042 : Result := Form_Utama.Lampu_2404b;
 11051 : Result := Form_Utama.Lampu_2405a;
 11052 : Result := Form_Utama.Lampu_2405b;
 11053 : Result := Form_Utama.Lampu_2405c;
 11054 : Result := Form_Utama.Lampu_2405d;
 1106 : Result := Form_Utama.Lampu_2406;
 1118 : Result := Form_Utama.Lampu_2418;
 1119 : Result := Form_Utama.Lampu_2419;

 12011 : Result := Form_Utama.Lampu_3401a;
 12012 : Result := Form_Utama.Lampu_3401b;
 12013 : Result := Form_Utama.Lampu_3401c;
 1202 : Result := Form_Utama.Lampu_3402;
 1203 : Result := Form_Utama.Lampu_3403;
 1204 : Result := Form_Utama.Lampu_3404;
 1205 : Result := Form_Utama.Lampu_3405;
 1218 : Result := Form_Utama.Lampu_3418;

```

    1219 : Result := Form_Utama.Lampu_3419;
Else
    Result := Form_Utama.Lampu_Kosong;
End;
End;

Procedure TForm_Utama.Gambar_Lampu(Nomor_Panel:Integer;Data : Integer;Status : Byte);
Var
    Posisi : Byte;
    Nomor_Lampu,N,M : Integer;
Begin
    Case Data Of
        1: Begin
            Posisi := $80;
            For N := 1 To 7 Do
                Begin
                    Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + FormatFloat('00',N));
                    Posisi := Posisi Shr 1;
                    IF Status AND Posisi <> 0
                    Then
                        Begin
                            IF D1_Panel[Nomor_Panel] AND Posisi = 0
                            Then
                                Begin
                                    IF jumlah_Lampu[Nomor_Lampu] = 1
                                    Then
                                        Begin
                                            Try
                                                Lampu(Nomor_Lampu).Visible := True;
                                            Except
                                                End;
                                            End
                                        End
                                    Else
                                        IF jumlah_Lampu[Nomor_Lampu] > 1 Then
                                            For M := 1 To jumlah_Lampu[Nomor_Lampu] DO
                                                Begin
                                                    Try
                                                        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
                                                    Except
                                                        End;
                                                    End;
                                                End;
                                            End;
                                        End;
                                    End
                                End
                            End
                        End
                    End
                End
            End
        End
    Else
        Begin
            IF D1_Panel[Nomor_Panel] AND Posisi <> 0
            Then
                Begin
                    IF jumlah_Lampu[Nomor_Lampu] = 1
                    Then
                        Begin
                            Try
                                Lampu(Nomor_Lampu).Visible := True;
                            Except
                                End;
                            End
                        End
                    End
                End
            End
        End
    End
End;

```

```

        Lampu(Nomor_Lampu).Visible := False;
    Except
    End;
End
Else
    IF jumlah_Lampu[Nomor_Lampu] > 1 Then
    For M := 1 To jumlah_Lampu[Nomor_Lampu] DO
        Begin
            Try
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
            Except
            End;
        End;
    End;
End;
End;
End;
2: Begin
    Posisi := $80;
    For N := 1 To 5 Do
        Begin
            Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + FormatFloat('00',N+7));
            Posisi := Posisi Shr 1;
            IF Status AND Posisi <> 0
            Then
                Begin
                    IF D2_Panel[RadioGroup1.ItemIndex+1] AND Posisi = 0
                    Then
                        Begin
                            IF jumlah_Lampu[Nomor_Lampu] = 1
                            Then
                                Begin
                                    Try
                                        Lampu(Nomor_Lampu).Visible := True;
                                    Except
                                    End;
                                End
                            Else
                                IF jumlah_Lampu[Nomor_Lampu] > 1 Then
                                For M := 1 To jumlah_Lampu[Nomor_Lampu] DO
                                    Begin
                                        Try
                                            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
                                        Except
                                        End;
                                    End;
                                End;
                            End;
                        End;
                    End
                End
            Else
                Begin
                    IF D2_Panel[RadioGroup1.ItemIndex-1] AND Posisi <> 0
                    Then

```

```

Begin
  IF jumlah_Lampu[Nomor_Lampu] = 1
  Then
    Begin
      Try
        Lampu(Nomor_Lampu).Visible := False;
      Except
      End;
    End
  Else
    IF jumlah_Lampu[Nomor_Lampu] > 1 Then
      For M := 1 To jumlah_Lampu[Nomor_Lampu] DO
        Begin
          Try
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
          Except
          End;
        End;
      End;
    End;
  End;
End;

Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + '16');
IF jumlah_Lampu[Nomor_Lampu] >= 1
Then
  Begin
    Posisi := $03;
    IF (status AND Posisi) = 0
    Then
      Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
          Begin
            Try
              Lampu(Nomor_Lampu).Picture := TL_Off_Off.Picture;
            Except
            End;
          End
        Else
          Begin
            Try
              For M:= 1 To 4 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
              For M:= 5 To 8 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
            Except
            End;
          End;
        End
      End
    Else
      IF (status AND Posisi) = 2
      Then

```

```

Begin
  IF jumlah_Lampu[Nomor_Lampu] = 1
  Then
    Begin
      Try
        Lampu(Nomor_Lampu).Picture := TL_On_Off.Picture;
      Except
        End;
      End
    Else
      Begin
        Try
          For M:= 1 To 4 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
          For M:= 5 To 8 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
          Except
            End;
          End;
        End;
      End
    Else
      Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
          Begin
            Try
              Lampu(Nomor_Lampu).Picture := TL_On_On.Picture;
            Except
              End;
            End
          Else
            Begin
              Try
                For M:= 1 To 4 Do
                  Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
                For M:= 5 To 8 Do
                  Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
                Except
                  End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  3: Begin
    For N := 1 To 3 Do
      Begin
        Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + FormatFloat('00',N+16));
        Posisi := $C0;
        Posisi := Posisi Shr (2*N-1);

```



```

IF (status AND Posisi) = 0
Then
  Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
      Begin
        Try
          Lampu(Nomor_Lampu).Picture := TL_Off_Off.Picture;
        Except
          End;
        End
      End
    Else
      Begin
        Try
          For M:= 1 To 4 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
          For M:= 5 To 8 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
          Except
            End;
          End;
        End
      End
    Else
      IF (status AND Posisi) Shr (7-2*(N)) = 2
      Then
        Begin
          IF jumlah_Lampu[Nomor_Lampu] = 1
          Then
            Begin
              Try
                Lampu(Nomor_Lampu).Picture := TL_On_Off.Picture;
              Except
                End;
              End
            End
          Else
            Begin
              Try
                For M:= 1 To 4 Do
                  Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
                For M:= 5 To 8 Do
                  Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := False;
                Except
                  End;
                End;
              End;
            End
          End
        End
      Else
        Begin
          IF jumlah_Lampu[Nomor_Lampu] = 1
          Then
            Begin
              Try

```

```

        Lampu(Nomor_Lampu).Picture := TL_On_On.Picture;
    except
    End;
End
Else
Begin
    Try
        For M:= 1 To 4 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
        For M:= 5 To 8 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(M))).Visible := True;
        Except
        End;
    End;
End;
End;
End;
Lampu_Kosong.Visible := False;
End;

```

```

Procedure TForm_Utama.Cek_Terima(datanya : String);
Var
    D1,D2,D3,
    D1_Inv,D2_Inv,D3_Inv : Byte;
    Sinar : real;
Begin
    Timer1.Enabled := False;
    IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
    IF CheckBox1.Checked Then Memo1.Lines.Add(datanya);
    D1 := Hexbin(Copy(datanya,1,2));
    D2 := Hexbin(Copy(datanya,3,2));
    D3 := Hexbin(Copy(datanya,5,2));
    D1_Inv := Hexbin(Copy(datanya,7,2));
    D2_Inv := Hexbin(Copy(datanya,9,2));
    D3_Inv := Hexbin(Copy(datanya,11,2));
    IF ( ((not(D1) AND $7F) = D1_Inv) AND
        ((not(D2) AND $7F) = D2_Inv) AND
        ((not(D3) AND $7F) = D3_Inv) )
    Then
        Begin
            panel_Error[RadioGroup1.ItemIndex+1] := 0;

            if RadioGroup1.ItemIndex = 12
            Then
                Begin
                    IF copy(datanya,1,2) = '3B'
                    Then Sinar := 0
                    Else IF copy(datanya,1,2) = '2A'
                    Then sinar := 9999
                    Else IF copy(datanya,1,2) = '7F'
                    Then sinar := 0
                End
            End
        End
    End

```

```

Else
  Begin
    Sinar := StrToInt(copy(datanya,1,2) + datanya[4] + datanya[6]);
    Case StrToInt(datanya[3]) of
      0 : Sinar := Sinar;
      1 : Sinar := Sinar/10;
      2 : Sinar := Sinar/100;
      3 : Sinar := Sinar*10;
      4 : Sinar := Sinar*100;
    End;
  End;

Label24.Caption := FloatToStr(Sinar);

IF Sinar < B_Sedang
Then Cahaya := 0
Else IF Sinar < B_Terang
  Then Cahaya := 1
  Else Cahaya := 2;

End
Else
  Begin
    IF D1 <> D1_Panel[RadioGroup1.ItemIndex+1]
    Then
      Begin
        Gambar_Lampu(RadioGroup1.ItemIndex+1,1,D1);
        D1_Panel[RadioGroup1.ItemIndex+1] := D1;
      End;
    IF D2 <> D2_Panel[RadioGroup1.ItemIndex+1]
    Then
      Begin
        Gambar_Lampu(RadioGroup1.ItemIndex+1,2,D2);
        D2_Panel[RadioGroup1.ItemIndex+1] := D2;
      End;
    IF D3 <> D3_Panel[RadioGroup1.ItemIndex+1]
    Then
      Begin
        Gambar_Lampu(RadioGroup1.ItemIndex+1,3,D3);
        D3_Panel[RadioGroup1.ItemIndex+1] := D3;
      End;
    End;
  End;
  Timer1.Enabled := True;
End;

procedure TForm_Utama.Satu_Relay(Nomor_Panel : byte; Nomor : Integer);
Var
  Kirim_Perintah : Boolean;
  Nomor_Perintah : Byte;
  Nomor_Lampu : Integer;
  Posisi : Byte;

```

```

Perintah_Panel : String;
N : Integer;
begin
    Kirim_Perintah := False;
    Nomor_Perintah := Nomor_Panel + 128;
    Perintah_Panel := DecToHex(Nomor_Perintah);
    Perintah_Panel := Perintah_Panel + DecToHex(not(Nomor_Perintah) OR $80);
    Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + Format('00', Nomor));

    Posisi := $80;

    IF Nomor <= 7
    Then
        Begin
            Posisi := Posisi Shr Nomor;

            IF (D1_Panel[Nomor_Panel] AND Posisi) = 0
            Then
                Begin
                    IF Cahaya <> 2 Then
                        Begin
                            IF jumlah_Lampu[Nomor_Lampu] = 1
                            Then
                                Lampu(Nomor_Lampu).Visible := True;
                            Else
                                For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
                                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := True;
                                D1_Panel[Nomor_Panel] :=
                                    D1_Panel[Nomor_Panel] OR Posisi;
                                Kirim_Perintah := True;
                            End;
                        End
                    Else
                        Begin
                            IF jumlah_Lampu[Nomor_Lampu] = 1
                            Then
                                Lampu(Nomor_Lampu).Visible := False;
                            Else
                                For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
                                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := False;
                                D1_Panel[Nomor_Panel] :=
                                    D1_Panel[Nomor_Panel] AND not(Posisi);
                                Kirim_Perintah := True;
                            End;
                        End
                    End
                Else
                    Begin
                        Posisi := Posisi Shr (Nomor-7);

                        IF (D2_Panel[Nomor_Panel] AND Posisi) = 0
                        Then
                            Begin

```

```

IF Cahaya <> 2 Then
Begin
  IF jumlah_Lampu[Nomor_Lampu] = 1
  Then
    Lampu(Nomor_Lampu).Visible := True
  Else
    For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
      Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := True;
    D2_Panel[Nomor_Panel] :=
      D2_Panel[Nomor_Panel] OR Posisi;
    Kirim_Perintah := True;
  End;
End
Else
Begin
  IF jumlah_Lampu[Nomor_Lampu] = 1
  Then
    Lampu(Nomor_Lampu).Visible := False
  Else
    For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
      Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := False;
    D2_Panel[Nomor_Panel] :=
      D2_Panel[Nomor_Panel] AND not(Posisi);
    Kirim_Perintah := True;
  End;
End;

IF Kirim_Perintah Then
Begin
  Kirim( Perintah_Panel +
    DecToHex(D1_Panel[Nomor_Panel]) +
    DecToHex(D2_Panel[Nomor_Panel]) +
    DecToHex(D3_Panel[Nomor_Panel]) +
    DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
    DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
    DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );

  mem01.Lines.Add(
    Perintah_Panel +
    DecToHex(D1_Panel[Nomor_Panel]) +
    DecToHex(D2_Panel[Nomor_Panel]) +
    DecToHex(D3_Panel[Nomor_Panel]) +
    DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
    DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
    DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );
  End;
end;

procedure TForm_Utama.Dua_Relay(Nomor_Panel : byte; Nomor : Integer);
Var
  Kirim_Perintah : Boolean;
  Nomor_Perintah : Byte;

```

```

Nomor_Lampu : Integer;
Posisi : Byte;
Perintah_Panel : String;
N : Integer;
begin
    Kirim_Perintah := False;
    Nomor_Perintah := Nomor_Panel + 128;
    Perintah_Panel := DecToHex(Nomor_Perintah);
    Perintah_Panel := Perintah_Panel + DecToHex(not(Nomor_Perintah) OR $80);
    Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + FormatFloat('00', Nomor));

    IF Nomor > 16
    Then
        Begin
            Posisi := $C0;
            Posisi := Posisi Shr (2*(Nomor-16)-1);

            IF (D3_Panel[Nomor_Panel] AND Posisi) = 0
            Then
                Begin
                    Case Cahaya OF
                        1: Begin
                            Posisi := $80;
                            Posisi := Posisi Shr (2*(Nomor-16)-1);

                            IF jumlah_Lampu[Nomor_Lampu] = 1
                            Then
                                Lampu(Nomor_Lampu).Picture := TL_On_Off.Picture
                            Else
                                Begin
                                    For N:= 1 To 4 Do
                                        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
                                    For N:= 5 To 8 Do
                                        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
                                    End;
                                    D3_Panel[Nomor_Panel] :=
                                        D3_Panel[Nomor_Panel] OR Posisi;
                                    Kirim_Perintah := True;
                                End;
                        0: Begin
                            IF jumlah_Lampu[Nomor_Lampu] = 1
                            Then
                                Lampu(Nomor_Lampu).Picture := TL_On_On.Picture
                            Else
                                Begin
                                    For N:= 1 To 4 Do
                                        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
                                    For N:= 5 To 8 Do
                                        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
                                    End;
                                    D3_Panel[Nomor_Panel] :=
                                        D3_Panel[Nomor_Panel] OR Posisi;

```

```

        Kirim_Perintah := True;
    End;
3: Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
        Lampu(Nomor_Lampu).Picture := TL_On_On.Picture
    Else
        Begin
            For N:= 1 To 4 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
            For N:= 5 To 8 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
            End;
            D3_Panel[Nomor_Panel] :=
                D3_Panel[Nomor_Panel] OR Posisi;
            Kirim_Perintah := True;
        End;

    End;
End
Else
    Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
            Lampu(Nomor_Lampu).Picture := TL_Off_Off.Picture
        Else
            Begin
                For N:= 1 To 4 Do
                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
                For N:= 5 To 8 Do
                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
                End;
                D3_Panel[Nomor_Panel] :=
                    D3_Panel[Nomor_Panel] AND not(Posisi);
                Kirim_Perintah := True;
            End;

        End
    End
Else
    Begin
        Posisi := $03;

        IF (D2_Panel[Nomor_Panel] AND Posisi) = 0
        Then
            Case Cahaya Of
            1: Begin
                IF jumlah_Lampu[Nomor_Lampu] = 1
                Then
                    Lampu(Nomor_Lampu).Picture := TL_On_Off.Picture
                Else
                    Begin
                        For N:= 1 To 4 Do

```

```

        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := True;
    For N:= 5 To 8 Do
        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := False;
    End;
    Posisi := $02;
    D2_Panel[Nomor_Panel] :=
        D2_Panel[Nomor_Panel] OR Posisi;
    Kirim_Perintah := True;
End;
0: Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
        Lampu(Nomor_Lampu).Picture := TL_On_On.Picture
    Else
        Begin
            For N:= 1 To 4 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := True;
            For N:= 5 To 8 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := True;
            End;
            D2_Panel[Nomor_Panel] :=
                D2_Panel[Nomor_Panel] OR Posisi;
            Kirim_Perintah := True;
        End;
3: Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
        Lampu(Nomor_Lampu).Picture := TL_On_On.Picture
    Else
        Begin
            For N:= 1 To 4 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := True;
            For N:= 5 To 8 Do
                Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := True;
            End;
            D2_Panel[Nomor_Panel] :=
                D2_Panel[Nomor_Panel] OR Posisi;
            Kirim_Perintah := True;
        End;
End;
Else
    Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
            Lampu(Nomor_Lampu).Picture := TL_Off_Off.Picture
        Else
            Begin
                For N:= 1 To 4 Do
                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := False;
                For N:= 5 To 8 Do
                    Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))). Visible := False;
                End;

```



```

        D2_Panel[Nomor_Panel] :=
            D2_Panel[Nomor_Panel] AND not(Posisi);
        Kirim_Perintah := True;
    End;
End;

IF Kirim_Perintah Then
Begin
    Kirim( Perintah_Panel +
        DecToHex(D1_Panel[Nomor_Panel]) +
        DecToHex(D2_Panel[Nomor_Panel]) +
        DecToHex(D3_Panel[Nomor_Panel]) +
        DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );

    memo1.Lines.Add(
        Perintah_Panel +
        DecToHex(D1_Panel[Nomor_Panel]) +
        DecToHex(D2_Panel[Nomor_Panel]) +
        DecToHex(D3_Panel[Nomor_Panel]) +
        DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );
    End;
End;

procedure TForm_Utama.Timer_Satu_Relay(Nomor_Panel : byte; Nomor : Integer; Status :
Integer);
Var
    Kirim_Perintah : Boolean;
    Nomor_Perintah : Byte;
    Nomor_Lampu : Integer;
    Posisi : Byte;
    Perintah_Panel : String;
    N : Integer;
begin
    Kirim_Perintah := False;
    Nomor_Perintah := Nomor_Panel + 128;
    Perintah_Panel := DecToHex(Nomor_Perintah);
    Perintah_Panel := Perintah_Panel + DecToHex(not(Nomor_Perintah) OR $80);
    Nomor_Lampu := StrToInt(IntToStr(Nomor_Panel) + FormatFloat('00', Nomor));

    Posisi := $80;

    IF Nomor <= 7
    Then
        Begin
            Posisi := Posisi Shr Nomor;

            IF Status = 1
            Then

```

```

Begin
  IF jumlah_Lampu[Nomor_Lampu] = 1
  Then
    Lampu(Nomor_Lampu).Visible := True
  Else
    For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
      Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := True;
    D1_Panel[Nomor_Panel] :=
      D1_Panel[Nomor_Panel] OR Posisi;
    Kirim_Perintah := True;
  End
Else
  Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
      Lampu(Nomor_Lampu).Visible := False
    Else
      For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
        Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := False;
      D1_Panel[Nomor_Panel] :=
        D1_Panel[Nomor_Panel] AND not(Posisi);
      Kirim_Perintah := True;
    End;
  End
Else
  Begin
    Posisi := Posisi Shr (Nomor-7);

    IF Status = 1
    Then
      Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
          Lampu(Nomor_Lampu).Visible := True
        Else
          For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := True;
          D2_Panel[Nomor_Panel] :=
            D2_Panel[Nomor_Panel] OR Posisi;
          Kirim_Perintah := True;
        End
      End
    Else
      Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
          Lampu(Nomor_Lampu).Visible := False
        Else
          For N := 1 To jumlah_Lampu[Nomor_Lampu] DO
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(n))).Visible := False;
          D2_Panel[Nomor_Panel] :=
            D2_Panel[Nomor_Panel] AND not(Posisi);
          Kirim_Perintah := True;
        End
      End
    End
  End

```

```

Else
  Begin
    For N:= 1 To 4 Do
      Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
    For N:= 5 To 8 Do
      Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
    End;
    D3_Panel[Nomor_Panel] :=
      D3_Panel[Nomor_Panel] OR Posisi;
    Kirim_Perintah := True;
  End
Else
  Begin
    IF jumlah_Lampu[Nomor_Lampu] = 1
    Then
      Lampu(Nomor_Lampu).Picture := TL_Off_Off.Picture
    Else
      Begin
        For N:= 1 To 4 Do
          Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
        For N:= 5 To 8 Do
          Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
        End;
        D3_Panel[Nomor_Panel] :=
          D3_Panel[Nomor_Panel] AND not(Posisi);
        Kirim_Perintah := True;
      End;
    End
Else
  Begin
    Posisi := $03;

    IF Status = 1
    Then
      Begin
        IF jumlah_Lampu[Nomor_Lampu] = 1
        Then
          Lampu(Nomor_Lampu).Picture := TL_On_On.Picture
        Else
          Begin
            For N:= 1 To 4 Do
              Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
            For N:= 5 To 8 Do
              Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := True;
            End;
            D2_Panel[Nomor_Panel] :=
              D2_Panel[Nomor_Panel] OR Posisi;
            Kirim_Perintah := True;
          End
        Else
          Begin
            IF jumlah_Lampu[Nomor_Lampu] = 1

```

```

Then
    Lampu(Nomor_Lampu).Picture := TI_Off_Off.Picture
Else
    Begin
        For N:= 1 To 4 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
        For N:= 5 To 8 Do
            Lampu(StrToInt(IntToStr(Nomor_Lampu)+IntToStr(N))).Visible := False;
        End;
        D2_Panel[Nomor_Panel] :=
            D2_Panel[Nomor_Panel] AND not(Posisi);
        Kirim_Perintah := True;
    End;
End;

IF Kirim_Perintah Then
Begin
    Kirim( Perintah_Panel +
        DecToHex(D1_Panel[Nomor_Panel]) +
        DecToHex(D2_Panel[Nomor_Panel]) +
        DecToHex(D3_Panel[Nomor_Panel]) +
        DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );

    memo1.Lines.Add(
        Perintah_Panel +
        DecToHex(D1_Panel[Nomor_Panel]) +
        DecToHex(D2_Panel[Nomor_Panel]) +
        DecToHex(D3_Panel[Nomor_Panel]) +
        DecToHex(not(D1_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D2_Panel[Nomor_Panel]) AND $7F) +
        DecToHex(not(D3_Panel[Nomor_Panel]) AND $7F) );
    End;
End;

procedure TForm_Utama.FormCreate(Sender: TObject);
Var
    N : Integer;
begin
    awal := 0;

    Direktoriya := GetCurrentDir + '\';

    ADOConnection1.ConnectionString :=
        'Provider=Microsoft.Jet.OLEDB.4.0;' +
        'Data Source=' + Direktoriya + 'Smart.mdb;' +
        'Persist Security Info=False';
    ADOConnection1.Connected;

    Cahaya := 0;

```

Display_Error := False;

For N := 1 To 13 Do
 panel_Error[N] := 0;

jumlah_Lampu[101] := 1;
jumlah_Lampu[102] := 1;
jumlah_Lampu[103] := 1;
jumlah_Lampu[104] := 1;
jumlah_Lampu[105] := 2;
jumlah_Lampu[106] := 3;
jumlah_Lampu[107] := 1;
jumlah_Lampu[108] := 0;
jumlah_Lampu[109] := 0;
jumlah_Lampu[110] := 0;
jumlah_Lampu[111] := 0;
jumlah_Lampu[112] := 0;
jumlah_Lampu[116] := 0;
jumlah_Lampu[117] := 2;
jumlah_Lampu[118] := 2;
jumlah_Lampu[119] := 1;

jumlah_Lampu[201] := 1;
jumlah_Lampu[202] := 1;
jumlah_Lampu[203] := 3;
jumlah_Lampu[204] := 1;
jumlah_Lampu[205] := 2;
jumlah_Lampu[206] := 1;
jumlah_Lampu[207] := 1;
jumlah_Lampu[208] := 1;
jumlah_Lampu[209] := 1;
jumlah_Lampu[210] := 1;
jumlah_Lampu[211] := 0;
jumlah_Lampu[212] := 0;
jumlah_Lampu[216] := 0;
jumlah_Lampu[217] := 0;
jumlah_Lampu[218] := 1;
jumlah_Lampu[219] := 1;

jumlah_Lampu[301] := 1;
jumlah_Lampu[302] := 1;
jumlah_Lampu[303] := 1;
jumlah_Lampu[304] := 2;
jumlah_Lampu[305] := 1;
jumlah_Lampu[306] := 2;
jumlah_Lampu[307] := 0;
jumlah_Lampu[308] := 0;
jumlah_Lampu[309] := 0;
jumlah_Lampu[310] := 0;
jumlah_Lampu[311] := 0;
jumlah_Lampu[312] := 0;
jumlah_Lampu[316] := 0;

jumlah_Lampu[317] := 0;
jumlah_Lampu[318] := 1;
jumlah_Lampu[319] := 1;

jumlah_Lampu[401] := 1;
jumlah_Lampu[402] := 1;
jumlah_Lampu[403] := 2;
jumlah_Lampu[404] := 5;
jumlah_Lampu[405] := 1;
jumlah_Lampu[406] := 1;
jumlah_Lampu[407] := 0;
jumlah_Lampu[408] := 0;
jumlah_Lampu[409] := 0;
jumlah_Lampu[410] := 0;
jumlah_Lampu[411] := 0;
jumlah_Lampu[412] := 0;
jumlah_Lampu[416] := 0;
jumlah_Lampu[417] := 0;
jumlah_Lampu[418] := 1;
jumlah_Lampu[419] := 1;

jumlah_Lampu[501] := 1;
jumlah_Lampu[502] := 1;
jumlah_Lampu[503] := 4;
jumlah_Lampu[504] := 1;
jumlah_Lampu[505] := 1;
jumlah_Lampu[506] := 1;
jumlah_Lampu[507] := 1;
jumlah_Lampu[508] := 1;
jumlah_Lampu[509] := 1;
jumlah_Lampu[510] := 1;
jumlah_Lampu[511] := 2;
jumlah_Lampu[512] := 1;
jumlah_Lampu[516] := 0;
jumlah_Lampu[517] := 1;
jumlah_Lampu[518] := 1;
jumlah_Lampu[519] := 1;

jumlah_Lampu[601] := 1;
jumlah_Lampu[602] := 1;
jumlah_Lampu[603] := 1;
jumlah_Lampu[604] := 1;
jumlah_Lampu[605] := 1;
jumlah_Lampu[606] := 1;
jumlah_Lampu[607] := 1;
jumlah_Lampu[608] := 0;
jumlah_Lampu[609] := 0;
jumlah_Lampu[610] := 0;
jumlah_Lampu[611] := 0;
jumlah_Lampu[612] := 0;
jumlah_Lampu[616] := 0;
jumlah_Lampu[617] := 0;

jumlah_Lampu[618] := 0;
jumlah_Lampu[619] := 1;

jumlah_Lampu[701] := 1;
jumlah_Lampu[702] := 1;
jumlah_Lampu[703] := 2;
jumlah_Lampu[704] := 1;
jumlah_Lampu[705] := 1;
jumlah_Lampu[706] := 0;
jumlah_Lampu[707] := 0;
jumlah_Lampu[708] := 0;
jumlah_Lampu[709] := 0;
jumlah_Lampu[710] := 0;
jumlah_Lampu[711] := 0;
jumlah_Lampu[712] := 0;
jumlah_Lampu[716] := 0;
jumlah_Lampu[717] := 1;
jumlah_Lampu[718] := 1;
jumlah_Lampu[719] := 0;

jumlah_Lampu[801] := 1;
jumlah_Lampu[802] := 1;
jumlah_Lampu[803] := 1;
jumlah_Lampu[804] := 4;
jumlah_Lampu[805] := 1;
jumlah_Lampu[806] := 1;
jumlah_Lampu[807] := 1;
jumlah_Lampu[808] := 1;
jumlah_Lampu[809] := 1;
jumlah_Lampu[810] := 0;
jumlah_Lampu[811] := 0;
jumlah_Lampu[812] := 0;
jumlah_Lampu[816] := 0;
jumlah_Lampu[817] := 0;
jumlah_Lampu[818] := 1;
jumlah_Lampu[819] := 1;

jumlah_Lampu[901] := 1;
jumlah_Lampu[902] := 1;
jumlah_Lampu[903] := 1;
jumlah_Lampu[904] := 1;
jumlah_Lampu[905] := 1;
jumlah_Lampu[906] := 1;
jumlah_Lampu[907] := 1;
jumlah_Lampu[908] := 2;
jumlah_Lampu[909] := 3;
jumlah_Lampu[910] := 4;
jumlah_Lampu[911] := 1;
jumlah_Lampu[912] := 1;
jumlah_Lampu[916] := 1;
jumlah_Lampu[917] := 1;
jumlah_Lampu[918] := 1;

jumlah_Lampu[919] := 1;

jumlah_Lampu[1001] := 3;
jumlah_Lampu[1002] := 5;
jumlah_Lampu[1003] := 0;
jumlah_Lampu[1004] := 0;
jumlah_Lampu[1005] := 0;
jumlah_Lampu[1006] := 0;
jumlah_Lampu[1007] := 0;
jumlah_Lampu[1008] := 0;
jumlah_Lampu[1009] := 0;
jumlah_Lampu[1010] := 0;
jumlah_Lampu[1011] := 0;
jumlah_Lampu[1012] := 0;
jumlah_Lampu[1016] := 0;
jumlah_Lampu[1017] := 1;
jumlah_Lampu[1018] := 1;
jumlah_Lampu[1019] := 1;

jumlah_Lampu[1101] := 1;
jumlah_Lampu[1102] := 1;
jumlah_Lampu[1103] := 1;
jumlah_Lampu[1104] := 2;
jumlah_Lampu[1105] := 4;
jumlah_Lampu[1106] := 1;
jumlah_Lampu[1107] := 0;
jumlah_Lampu[1108] := 0;
jumlah_Lampu[1109] := 0;
jumlah_Lampu[1110] := 0;
jumlah_Lampu[1111] := 0;
jumlah_Lampu[1112] := 0;
jumlah_Lampu[1116] := 0;
jumlah_Lampu[1117] := 0;
jumlah_Lampu[1118] := 1;
jumlah_Lampu[1119] := 1;

jumlah_Lampu[1201] := 3;
jumlah_Lampu[1202] := 1;
jumlah_Lampu[1203] := 1;
jumlah_Lampu[1204] := 1;
jumlah_Lampu[1205] := 1;
jumlah_Lampu[1206] := 0;
jumlah_Lampu[1207] := 0;
jumlah_Lampu[1208] := 0;
jumlah_Lampu[1209] := 0;
jumlah_Lampu[1210] := 0;
jumlah_Lampu[1211] := 0;
jumlah_Lampu[1212] := 0;
jumlah_Lampu[1216] := 0;
jumlah_Lampu[1217] := 0;
jumlah_Lampu[1218] := 1;
jumlah_Lampu[1219] := 1;

```

For N := 1 To 12 Do
Begin
    D1_Panel[N] := 0;
    D2_Panel[N] := 0;
    D3_Panel[N] := 0;
End;

panel[11] := 1;
panel[21] := 2;
panel[31] := 3;
panel[12] := 4;
panel[22] := 5;
panel[32] := 6;
panel[13] := 7;
panel[23] := 8;
panel[33] := 9;
panel[14] := 10;
panel[24] := 11;
panel[34] := 12;

ADOTable1.Active := True;
ADOTable2.Active := True;

For N:= 1 To ADOTable1.RecordCount Do
Begin
    Mati[N] := ADOTable1.Mati.Value;
    Hidup[N] := ADOTable1.Hidup.Value;
    ADOTable1.Next;
End;

ADOTable2.First;
B_Terang := ADOTable2.Batas_Bawah.Value;
ADOTable2.Next;
B_Sedang := ADOTable2.Batas_Bawah.Value;
Label22.Caption := 'Batas Bawah Cahaya ' + ADOTable2.Keterangan.Value ;
Edit10.Text := IntToStr(ADOTable2.Batas_Bawah.Value);

Try
    // VaComm1.Open;
    {Proses_Kirim := False;
    Operatornya := ";    }
    // Vacomm1.SetRTS(True); // direction => receive
Except
End;

end;

procedure TForm_Utama.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    IF Key = Chr(VK_RETURN)
    Then

```

```

    Begin
        Key := #0;
        SendMessage(Handle, WM_NEXTDLGCTL, 0, 0);
    End;
end;

procedure TForm_Utama.VaComm1RxChar(Sender: TObject; Count: Integer);
var
    I: Integer;
    Tmp,terima: string;
begin
    { Tmp := VaComm1.ReadText;

    for I := 1 to Length(Tmp) do
        Data_Terima := Data_Terima + DecToHex(ord(Tmp[I]));

    IF Length(Data_Terima) = 12
    Then
        Cck_Terima(Data_Terima);

        Data_Terima := ""; } //reset received message
end;

procedure TForm_Utama.Timer1Timer(Sender: TObject);
var
    Status_Panel,
    Status_Cahaya : String;
    nomor_Panel : Byte;
begin
    IF RadioGroup1.ItemIndex = 12
    Then
        Begin
            RadioGroup1.ItemIndex := 0;
            Case Cahaya of
                0: Status_Cahaya:='8EF10000007F7F7F';
                1: Status_Cahaya:='8EF10100007E7F7F';
                2: Status_Cahaya:='8EF10200007D7F7F';
                3: Status_Cahaya:='8EF10000007F7F7F';
            End;
            awal := 1;
        End
    Else
        if awal = 0
        then
            RadioGroup1.ItemIndex := RadioGroup1.ItemIndex + 1
        Else
            awal := 0;

    IF RadioGroup1.ItemIndex = 12

```



```

Then
  Begin
    Kirim('CEB1');
  End
Else
  Begin
    nomor_Panel := RadioGroup1.ItemIndex+193;
    Status_Panel := DecToHex(nomor_Panel);
    Status_Panel := Status_Panel + DecToHex(not(nomor_Panel) OR $80);

    Kirim(Status_Panel);
    ClientSocket1.SocketSendText(Status_Panel);
    IF Memo1.Lines.Count > 10 Then Memo1.Lines.Clear;
    IF CheckBox1.Checked Then memo1.Lines.Add(Status_Panel);

  End;

IF panel_Error[RadioGroup1.ItemIndex+1] < 5
Then
  panel_Error[RadioGroup1.ItemIndex+1] :=
    panel_Error[RadioGroup1.ItemIndex+1] + 1;
end;

procedure TForm_Utama.Button1Click(Sender: TObject);
begin
  Memo1.Lines.Clear;
end;

procedure TForm_Utama.Edit7KeyPress(Sender: TObject; var Key: Char);
begin
  IF not ( key in ['A'..'F','0'..'9','a'..'f'])
  Then
    Key := #0;
end;

procedure TForm_Utama.Tombol_3118MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  (Sender As TPanel).BevelInner := bvLowered;
  (Sender As TPanel).BevelOuter := bvLowered;
end;

procedure TForm_Utama.Tombol_3118MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  (Sender As TPanel).BevelInner := bvRaised;
  (Sender As TPanel).BevelOuter := bvRaised;
end;

procedure TForm_Utama.Tombol_1101Click(Sender: TObject);
Var
  No_Input : Integer;

```

```

    No_Panel : Byte;
    x,y,a: string;
begin
    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
    Satu_Relay(No_Panel,No_Input);
    x := inttostr(no_panel);
    y := inttostr(no_input);
    a := (x+', '+y);
    ClientSocket1.Socket.SendText(a);
end;

```

```

procedure TForm Utama.Tombol_1117Click(Sender: TObject);
Var
    No_Input : Integer;
    No_Panel : Byte;
    x,y,a,datanya : string;
begin
    No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
    No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
    Dua_Relay(No_Panel,No_Input);
    X := inttostr(no_panel);
    y := inttostr(no_input);
    a := (x+', '+y);
    ClientSocket1.Socket.SendText(a);
end;

```

```

procedure TForm Utama.Timer2Timer(Sender: TObject);
begin
    IF panel_Error[1] = 5
        Then Panel_11.Visible := Display_Error
        Else Panel_11.Visible := False;
    IF panel_Error[2] = 5
        Then Panel_21.Visible := Display_Error
        Else Panel_21.Visible := False;
    IF panel_Error[3] = 5
        Then Panel_31.Visible := Display_Error
        Else Panel_31.Visible := False;
    IF panel_Error[4] = 5
        Then Panel_12.Visible := Display_Error
        Else Panel_12.Visible := False;
    IF panel_Error[5] = 5
        Then Panel_22.Visible := Display_Error
        Else Panel_22.Visible := False;
    IF panel_Error[6] = 5
        Then Panel_32.Visible := Display_Error
        Else Panel_32.Visible := False;
    IF panel_Error[7] = 5
        Then Panel_13.Visible := Display_Error
        Else Panel_13.Visible := False;
    IF panel_Error[8] = 5
        Then Panel_23.Visible := Display_Error

```

```

    Else Panel_23.Visible := False;
  IF panel_Error[9] = 5
    Then Panel_33.Visible := Display_Error
    Else Panel_33.Visible := False;
  IF panel_Error[10] = 5
    Then Panel_14.Visible := Display_Error
    Else Panel_14.Visible := False;
  IF panel_Error[11] = 5
    Then Panel_24.Visible := Display_Error
    Else Panel_24.Visible := False;
  IF panel_Error[12] = 5
    Then Panel_34.Visible := Display_Error
    Else Panel_34.Visible := False;

  IF panel_Error[13] = 5
  Then
    Begin
      //
    End;

  Display_Error := Not (Display_Error);

end;

procedure TForm_Utama.Button2Click(Sender: TObject);
begin
  Cahaya := 0;
  ClientSocket1.Socket.SendText('c');
end;

procedure TForm_Utama.Button3Click(Sender: TObject);
begin
  Cahaya := 1;
  ClientSocket1.Socket.SendText('k');
end;

procedure TForm_Utama.Button4Click(Sender: TObject);
begin
  Cahaya := 2;
  ClientSocket1.Socket.SendText('o');
end;

```

```

procedure TForm_Utama.Tombol_1107Click(Sender: TObject);
Var
  No_Input : Integer;
  No_Panel : Byte;
  Status_Cahaya : Integer;
  a,x,y : string;
begin
  No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
  No_Input := StrToInt(Copy(IntToStr((Sender As TPanel).Tag),3,2));
  Status_Cahaya := Cahaya;
  Cahaya := 0;
  Satu_Relay(No_Panel,No_Input);
  Cahaya := Status_Cahaya;
  x := inttostr(no_panel);
  y := inttostr(no_input);
  a := (x+', '+y);
  {memo2.Lines.Add(x+y);
  edit13.Text := inttostr(No_input);}
  ClientSocket1.Socket.SendText(a);
end;

```

```

procedure TForm_Utama.Tombol_3118ContextPopup(Sender: TObject;
  MousePos: TPoint; var Handled: Boolean);
Var
  No_Input : Integer;
  No_Panel : Byte;
begin
  Panel3.BringToFront;
  Panel3.Visible := True;

  No_Panel := Panel[StrToInt(Copy(IntToStr((Sender As TPanel).Tag),1,2))];
  No_Input := (Sender As TPanel).Tag;

  IF ADOTable1.Locate('nomor',No_Input,[])
  Then
    Begin
      Edit7.Text := IntToStr(ADOTable1.Lantai.Value);
      Edit8.Text := ADOTable1.Nama.Value;
      Edit9.Text := IntToStr(ADOTable1.Nomor.Value);
      DateTimePicker1.Time := ADOTable1.Hidup.Value;
      DateTimePicker2.Time := ADOTable1.Mati.Value;
    End
  Else
    Begin
      DateTimePicker1.Time := Time;
      DateTimePicker2.Time := Time;
    End;

    IF No_Panel <=3 Then Edit7.Text := '1'
    Else IF No_Panel <=6 Then Edit7.Text := '2'
    Else IF No_Panel <=9 Then Edit7.Text := '3'
    Else Edit7.Text := '4';
    Edit8.Text := (Sender As TPanel).Caption;
    Edit9.Text := IntToStr(No_Input);
  end;

procedure TForm_Utama.BitBtn5Click(Sender: TObject);
Var
  N : Integer;
begin
  Timer4.Enabled := False;
  IF ADOTable1.Locate('nomor',StrToInt(Edit9.Text),[])
  Then
    with ADOTable1 Do
      Begin
        Edit;
        ADOTable1.Hidup.Value := DateTimePicker1.Time;
        ADOTable1.Mati.Value := DateTimePicker2.Time;
        Post;
      End
    Else

```

```

with ADOTable1 Do
Begin
    Edit;
    Insert;
    ADOTable1.Lantai.Value := StrToInt(Edit7.Text);
    ADOTable1.Nomor.Value := StrToInt(Edit9.Text);
    ADOTable1>Nama.Value := Edit8.Text;
    ADOTable1.Hidup.Value := DateTimePicker1.Time;
    ADOTable1.Mati.Value := DateTimePicker2.Time;
    Post;
End;
Panel3.Visible := False;

ADOTable1.First;
For N:= 1 To ADOTable1.RecordCount Do
Begin
    Mati[N] := ADOTable1.Mati.Value;
    Hidup[N] := ADOTable1.Hidup.Value;
    ADOTable1.Next;
End;

Timer4.Enabled := True;
end;

procedure TForm_Utama.BitBtn6Click(Sender: TObject);
begin
    Panel3.Visible := False;
end;

procedure TForm_Utama.Timer4Timer(Sender: TObject);
Var
    N : Integer;
    No_Input : Integer;
    No_Panel : Byte;
begin
    Label20.Caption := TimeToStr(Time);
    For N := 1 To ADOTable1.RecordCount Do
    Begin
        IF TimeToStr(Time) = TimeToStr(Mati[N])
        Then
            Begin
                ADOTable1.RecNo := N;
                No_Input := StrToInt(Copy(IntToStr(ADOTable1.Nomor.Value),3,2));
                No_Panel := Panel[StrToInt(Copy(IntToStr(ADOTable1.Nomor.Value),1,2))];
                IF No_Input >= 16
                Then
                    Timer_Dua_Relay(No_Panel,No_Input,0)
                Else
                    Timer_Satu_Relay(No_Panel,No_Input,0);
            End;
        End;
    End;

```

```

IF TimeToStr(Time) = TimeToStr(Hidup[N])
Then
  Begin
    ADOTable1.RecNo := N;
    No_Input := StrToInt(Copy(IntToStr(ADOTable1.Nomor.Value),3,2));
    No_Panel := Panel[StrToInt(Copy(IntToStr(ADOTable1.Nomor.Value),1,2))];
    IF No_Input >= 16
    Then
      Timer_Dua_Relay(No_Panel,No_Input,1)
    Else
      Timer_Satu_Relay(No_Panel,No_Input,1);
    End;
  End;
end;

procedure TForm_Utama.BitBtn7Click(Sender: TObject);
begin
  IF ADOTable1.RecordCount > 0
  Then
    Begin
      ADOTable1.Delete;
      Edit11.Clear;
      Edit12.Clear;
    End;
  end;

procedure TForm_Utama.Edit10KeyPress(Sender: TObject; var Key: Char);
begin
  IF not (key In ['0'..'9',#13,#8])
  Then
    Key := #0;
  end;

procedure TForm_Utama.DBGrid2CellClick(Column: TColumn);
begin
  Label22.Caption := 'Batas Bawah Cahaya ' + ADOTable2.Keterangan.Value ;
  Edit10.Text := IntToStr(ADOTable2.Batas_Bawah.Value);
  Edit10.SetFocus;
end;

procedure TForm_Utama.DBGrid1CellClick(Column: TColumn);
begin
  Edit11.Text := IntToStr(ADOTable1.Lantai.Value);
  Edit12.Text := ADOTable1.Nama.Value;
  DateTimePicker3.Time := ADOTable1.Hidup.Value;
  DateTimePicker4.Time := ADOTable1.Mati.Value;
  DateTimePicker3.SetFocus;
end;

procedure TForm_Utama.BitBtn8Click(Sender: TObject);
begin
  IF ADOTable1.Locate('nama',Edit12.Text,[])

```

```

Then
  Begin
    ADOTable1.Edit;
    ADOTable1Hidup.Value := DateTimePicker3.Time;
    ADOTable1Mati.Value := DateTimePicker4.Time;
    ADOTable1.Post;
  End;
end;

procedure TForm_Utama.BitBtn9Click(Sender: TObject);
begin
  IF StrToIntDef(Edit10.Text,0) <= 4000
  Then
    Begin
      ADOTable2.Edit;
      ADOTable2Batas_Bawah.Value := StrToIntDef(Edit10.Text,0);
      ADOTable2.Post;
      Edit10.Clear;
      ADOTable2.First;
      B_Terang := ADOTable2Batas_Bawah.Value;
      ADOTable2.Next;
      B_Sedang := ADOTable2Batas_Bawah.Value;
      Label22.Caption := 'Batas Bawah Cahaya ' + ADOTable2Keterangan.Value ;
      Edit10.Text := IntToStr(ADOTable2Batas_Bawah.Value);
    End
  Else
    MessageDlg('besarnya harus antara 0 sampai 4000',mtError,[mbOk],0);

end;

procedure TForm_Utama.BitBtn10Click(Sender: TObject);
begin
  WinExec('editor.exe',WS_MAXIMIZE);
end;

procedure TForm_Utama.CheckBox2Click(Sender: TObject);
begin
  if not ClientSocket1.Active then
    ClientSocket1.Address := alamat.Text;
    ClientSocket1.Active := CheckBox2.Checked;
  end;
end;

procedure TForm_Utama.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
Var x : string;
Begin
  x := Socket.ReceiveText;
  Cek_Terima(x);
  x := "";
End;

```



```

procedure TForm_Utama.ServerSocket1ClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Label31.Caption := 'connected';
end;

procedure TForm_Utama.ServerSocket1ClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  Label31.Caption := 'disconnected';
end;

procedure TForm_Utama.ClientClick(Sender: TObject);
begin
  IF Client.ItemIndex = 0
  Then
  begin
    ClientSocket1.Address :=
      InputBox('Setup Koneksi', 'IP Server', '');
    ClientSocket1.Active := True;
    Label30.Caption := 'Terhubung';
    Label30.Font.Color := clGreen;
  end
  Else
  Begin
    ClientSocket1.Active := False;
    Label30.Caption := 'Terputus';
    Label30.Font.Color := clRed;
  End;
end;

end.

```